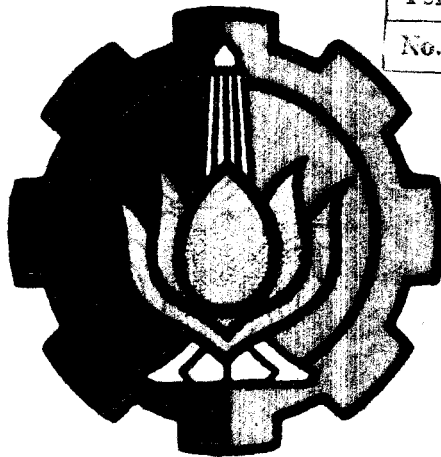


6170/TS/H/94

PERENCANAAN DAN PEMBUATAN
PROGRAMMABLE KONTROLER INTERFACE
UNTUK BUDIDAYA ANGGREK DALAM RUMAH KACA
YANG DIINTERFACEKAN KE IBM PC-XT

PERPUSTAKAAN I T S	
Tgl. Terima	
Terima Dari	
No. Agenda Prp.	



RSE
621.398.1
Neg
P-1
1993

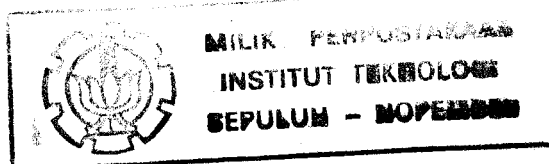
Oleh :

I MADE WIRA SATRIA NEGARA

NRP : 2882201077

JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA

1993



**PERENCANAAN DAN PEMBUATAN
PROGRAMMABLE KONTROLER INTERFACE
UNTUK BUDIDAYA ANGGREK DALAM RUMAH KACA
YANG DIINTERFACEKAN KE IBM PC-XT**

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik Elektro**

Pada

Bidang Studi Elektronika

Jurusan Teknik Elektro

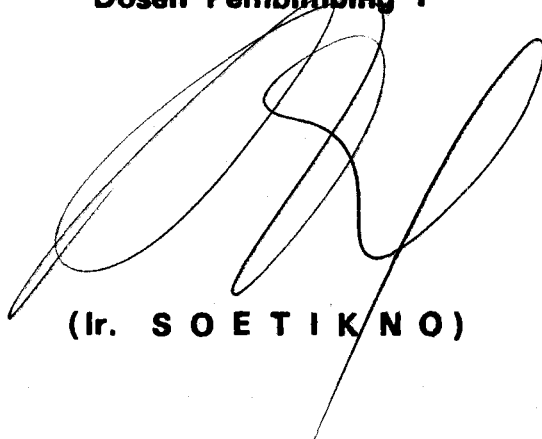
Fakultas Teknologi Industri

Institut Teknologi Sepuluh Nopember

S u r a b a y a

Mengetahui/Menyetujui

Dosen Pembimbing I



(Ir. S O E T I K N O)

Dosen Pembimbing II



(Ir. HENDRA KUSUMA)

**S U R A B A Y A
AGUSTUS, 1993**

ABSTRAK

Budi daya dan pemeliharaan Anggrek sekarang ini semakin meluas, tidak hanya oleh kalangan bermodal tetapi juga oleh kalangan rumah tangga. Tak dapat disangkal bahwa Anggrek merupakan bunga yang mempunyai nilai estetik dan ekonomi yang tinggi.

Keberhasilan dalam budidaya tentunya akan memberikan keuntungan dan kepuasan tersendiri. Karenanya perlu diperhatikan cara pemeliharaan yang baik meliputi pengaturan temperatur, kelembaban relatif, intensitas cahaya, kerap penyiraman dan waktu pemupukan untuk jenis Anggrek tertentu.

Untuk fleksibilitas dan otomatisasi dalam pemeliharaan, telah dipikirkan pembuatan suatu alat kontrol yang dapat diprogram disesuaikan dengan parameter-parameter di atas untuk jenis Anggrek yang berbeda-beda. Dalam hal pengukuran kelembaban relatif, ada dua alternatif konversi besaran yang digunakan yaitu berdasar tabel psycrometric dan rumus persamaan. Sedangkan intensitas cahaya maksimum yang dikontrol sebesar 1350 lux.

Pengkondisian dilakukan dalam suatu rumah kaca guna memudahkan pengontrolan, yang mana juga memberikan sudut pandang estetika yang baik dan dimungkinkan untuk pemeliharaan di dalam rumah.

KATA PENGANTAR

Puji syukur penyusun panjatkan ke hadirat Tuhan Yang Maha Esa atas rahmat dan karuniaNya sehingga dapat terselesaikannya Tugas Akhir dengan judul :

PERENCANAAN DAN PEMBUATAN PROGRAMMABLE KONTROLER INTERFACE UNTUK BUDI DAYA ANGGREK DALAM RUMAH KACA YANG DIINTERFACEKAN KE IBM PC/XT

Tugas akhir ini disusun guna memenuhi sebagian persyaratan untuk mencapai gelar kesarjanaan pada Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember di Surabaya.

Dalam mengerjakan tugas akhir ini penyusun banyak mendapatkan dukungan, bantuan, dan bimbingan dari berbagai pihak. Oleh karena itu pada kesempatan ini penyusun menyampaikan rasa terima kasih yang sebesar-besarnya kepada :

1. Ir. Soetikno, selaku Koordinator Bidang Studi Elektronika, Jurusan Teknik Elektro, Fakultas Teknologi Industri ITS, dosen pembimbing dan dosen wali.
2. Ir. Hendra Kusuma, selaku dosen pembimbing.
3. Ir. Katjuk Astrowulan, MSEE., selaku Ketua Jurusan Teknik Elektro FTI-ITS.

4. Papa, Mama, kakak dan adik-adik tercinta serta keluarga lainnya yang selalu memberikan dorongan moril maupun materiil sehingga penyusun dapat menyelesaikan studi.
5. Meriana, yang telah banyak memberikan bantuan dan dorongan moril kepada penyusun dalam menyelesaikan Tugas Akhir ini.
6. Rekan-rekan mahasiswa Jurusan Teknik Elektro FTI-ITS, khususnya bidang studi Elektronika, dan semua pihak yang telah memberikan dorongan dan bantuannya, baik secara langsung maupun tak langsung.

Akhir kata, penyusun berharap semoga apa yang telah dihasilkan dalam Tugas Akhir ini dapat dimanfaatkan untuk pengembangan keilmuan dan terapan aplikasinya.

Surabaya, Juli 1993

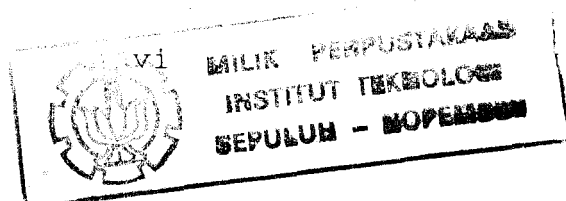
Penyusun

DAFTAR ISI

B A B	HALAMAN
ABSTRAK	i
KATA PENGANTAR	ii
DAFTAR ISI	iv
DAFTAR GAMBAR	ix
DAFTAR TABEL	xii
I. PENDAHULUAN	
I.1. LATAR BELAKANG	1
I.2. PERMASALAHAN	2
I.3. TUJUAN	2
I.4. METODOLOGI	3
I.5. SISTEMATIKA	3
I.6. RELEVANSI	4
II. TEORI PENUNJANG	
II.1. DESKRIPSI UMUM TANAMAN ANGGREK	5
II.1.1. MORFOLOGI DAN CARA HIDUP TANAMAN ANGGREK	5
II.1.2. KEBUTUHAN PRIMER TANAMAN ANGGREK	7
II.1.2.1. TEMPERATUR DAN KELEMBABAN	8
II.1.2.2. CAHAYA	8
II.1.2.3. PENYIRAMAN	10

II.1.2.4. PEMUPUKAN	13
II.1.2.5. ANGIN	13
II.1.2.6. PENGENDALIAN HAMA DAN PENYAKIT	14
II.2. AIR CONDITIONING SYSTEM	15
II.2.1. DESKRIPSI PSYCHROMETRIC CHART	15
II.2.2. METODA PENGGUNAAN PSYCHROMET- RIC CHART	18
II.2.3. HEATING DAN HUMIDIFYING	19
II.2.4. COOLING DAN DEHUMIDIFICATION	23
II.3. PERPINDAHAN KALOR	26
II.3.1. KONDUKSI	26
II.3.2. KONVEKSI	28
II.4. TRANSDUSER	30
II.4.1. SENSOR TEMPERATUR	30
II.4.2. SENSOR KELEMBABAN RELATIF	34
II.4.3. SENSOR CAHAYA	37
II.5. KONTROLER DIGITAL	41
II.5.1. KONTROL DUA POSISI/ON-OFF	42
II.5.2. KONTROL PROPORSIONAL PLUS INTEGRAL PLUS DERIVATIF	43
III. PERENCANAAN DAN PEMBUATAN	
III.1. UMUM	46
III.2. PERENCANAAN RUMAH KACA (GREEN HOUSE)	48
III.2.1. KONSTRUKSI RUMAH KACA	48

III.2.2. PERENCANAAN PEMANAS	50
III.3. PERENCANAAN HARDWARE	56
III.3.1. MODUL INTERFACE	56
III.3.1.1. RANGKAIAN BUFFER	57
III.3.1.2. RANGKAIAN DECODER	60
III.3.1.3. RANGKAIAN PPI 8255	64
III.3.1.4. RANGKAIAN PIT 8253	66
III.3.2. MODUL PENGINDRA SENSOR	70
III.3.2.1. RANGKAIAN MULTI- PLEXER DAN DE - MULTIPLEXER	70
III.3.2.2. RANGKAIAN DETEKTOR TEMPERATUR	73
III.3.2.3. RANGKAIAN DETEKTOR KELEMBABAN RELATIF	75
III.3.2.4. RANGKAIAN DETEKTOR INTENSITAS CAHAYA	76
III.3.3. MODUL ANALOG TO DIGITAL CON - VERTER	79
III.3.4. MODUL DRIVER	84
III.3.4.1. RANGKAIAN PENGAK - TIF POMPA PENGISAP	85
III.3.4.2. RANGKAIAN PENGGE - RAK MOTOR STEPPER	87
III.3.4.3. RANGKAIAN PENGGE -	



	RAK FAN	89
III.3.4.4.	RANGKAIAN PENG - ATUR PEMANAS	91
III.3.4.5.	RANGKAIAN DIMMER LAMPU FLOURESCENT	93
III.4.	PERENCANAAN SOFTWARE	96
III.4.1.	INISIALISASI PORT DAN TIMER	97
III.4.2.	INTERRUPT SERVICE ROUTINE ...	98
III.4.2.1.	PELAYANAN SETTING PERALATAN	99
III.4.2.2.	KONTROL UMPAN BALIK	101
III.4.2.3.	PELAYANAN SAMPLING PARAMETER	102
IV.	PENGUJIAN DAN PENGUKURAN	
IV.1.	PENGUJIAN SENSOR TEMPERATUR	109
IV.2.	PENGUJIAN SENSOR KELEMBABAN	110
IV.3.	PENGUJIAN SENSOR CAHAYA	113
IV.4.	PENGUJIAN TERHADAP RANGKAIAN TRIGGER ..	118
IV.5.	FASILITAS PADA PERANGKAT LUNAK	119
IV.6.	PERALATAN ELEKTRONIK DAN PROTOTYPE ALAT YANG DIBUAT	120
V.	PENUTUP	
V.1.	KESIMPULAN	121
V.2.	SARAN	122

DAFTAR PUSTAKA	123
----------------------	-----

LAMPIRAN

- A. PSYCHROMETRIC CHART, TABEL PSYCHROMETRIC DAN STEAM
TABLE
- B. LISTING PROGRAM
- C. RIWAYAT HIDUP

DAFTAR GAMBAR

GAMBAR	HALAMAN
2.1. METODA PENGGUNAAN PSYCHROMETRIC CHART ...	20
2.2. PROSES HEATING DAN HUMIDIFYING	22
2.3. PROSES COOLING DAN DEHUMIDIFICATION	24
2.4. IC SENSOR LM 335	33
2.5. PENGUKURAN KELEMBABAN RELATIF DENGAN PSYCHOMETER	37
2.6. NOTASI DAN KARAKTERISTIK PHOTOVOLTAIC CELL	40
2.7. DIAGRAM BLOK KONTROLER DUA POSISI	42
3.1. BLOK DIAGRAM PERANGKAT KERAS SISTEM	47
3.2. KONSTRUKSI MODEL RUMAH KACA	49
3.3. RANGKAIAN BUFFER PADA INTERFACE CARD	58
3.4. MODUL BUFFER I/O	59
3.5. DIAGRAM LOGIKA FUNGSI I/O DECODING DENGAN PAL 16L8	63
3.6. RANGKAIAN DEKODER DENGAN PAL 16L8	64
3.7. RANGKAIAN PPI 8255	65
3.8. RANGKAIAN PIT 8253	66
3.9. RANGKAIAN DAN SINYAL TRIGGER	68
3.10. RANGKAIAN CLOCK PIT 8253	70
3.11. RANGKAIAN MULTIPLEXER DAN DEMULTIPLEXER	72
3.12. RANGKAIAN DETEKTOR TEMPERATUR	75

3.13. RANGKAIAN DETEKTOR KELEMBABAN RELATIF ...	77
3.14. RANGKAIAN DETEKTOR INTENSITAS CAHAYA	79
3.15. BLOK DIAGRAM DAN DIAGRAM WAKTU ADC 0808 .	81
3.16. RANGKAIAN ADC	82
3.17. CLOCK ADC 0808	83
3.18. RANGKAIAN PENGHASIL TEGANGAN REFERENSI ..	84
3.19. RANGKAIAN PENGAKTIF POMPA PENGISAP	86
3.20. RANGKAIAN DRIVER MOTOR STEPPER	89
3.21. RANGKAIAN PENGGERAK FAN	91
3.22. RANGKAIAN PENGATUR PEMANAS	92
3.23. SIKLUS PENYALAN TRIAC	94
3.24. RANGKAIAN DIMMER LAMPU FLOURESCENT	95
3.25. DIAGRAM ALIR PERANGKAT LUNAK	96
3.26. ALUR KONTROL INTERRUPT SERVICE ROUTINE ..	99
3.27. DIAGRAM ALIR PELAYANAN PERALATAN	100
3.28. ALUR KONTROL BESARAN INTENSITAS CAHAYA ..	105
3.29. ALUR KONTROL BESARAN TEMPERATUR DAN INTENSITAS CAHAYA	108
4.1. PENYERAHAN TEGANGAN JALA-JALA DAN SINYAL TRIGGER (FOTO)	118
4.2 TAMPILAN FASILITAS PERANGKAT LUNAK (FOTO)	119
4.3. PERANGKAT KERAS PERALATAN YANG DIBUAT (FOTO)	120

DAFTAR TABEL

TABEL	HALAMAN
2-1. HASIL PROSES HEATING DAN HUMIDIFYING	23
2-2. HASIL PROSES COOLING DAN DEHUMIDIFICATION ..	25
2-3. PERBANDINGAN SENSOR TEMPERATUR ELEKTRIS	32
2-4. PERBANDINGAN SENSOR KELEMBABAN	35
2-5. PERBANDINGAN SENSOR CAHAYA	39
3-1. TABEL KEBENARAN PENGALAMATAN I/O PORT	62
3-2. FORMAT PEMILIHAN CHANNEL SAMPLING PARAMETER	103
4-1. HASIL PENGUJIAN SENSOR TEMPERATUR	110
4-2. HASIL PENGUJIAN SENSOR KELEMBABAN RELATIF ..	114
4-3. HUBUNGAN ARUS HUBUNG SINGKAT DENGAN INTEN- SITAS CAHAYA PADA SOLAR CELL	115
4-4. HASIL PENGUJIAN SENSOR CAHAYA	117

BAB I

PENDAHULUAN

I.1. LATAR BELAKANG

Budi daya anggrek saat ini merupakan salah satu agrobisnis yang mempunyai prospek yang cerah. Untuk mendapatkan hasil budi daya yang optimal diperlukan cara pemeliharaan yang benar, diantaranya dengan pengkondisian parameter-parameter seperti temperatur, kelembaban, intensitas cahaya, waktu penyiraman dan pemupukan yang sesuai dengan jenis anggrek tertentu.

Pembudidayaan anggrek biasanya dilakukan di alam terbuka, karenanya sangat bergantung pada kondisi iklim setempat. Untuk mengurangi ketergantungan pada iklim, pada beberapa tanaman budi daya, pembudidayaan dilakukan dalam suatu rumah kaca. Namun ada beberapa kekurangan pada rumah kaca yang telah ada yaitu :

- Pengontrolan dan pengkondisian ruangan masih dilakukan secara manual dengan alat-alat tertentu oleh tenaga manusia sepenuhnya sehingga fleksibilitasnya sangat terbatas dan belum ada otomatisasi.
- Belum ada konstruksi dan pengkondisian rumah kaca yang spesifik untuk keperluan budi daya anggrek.

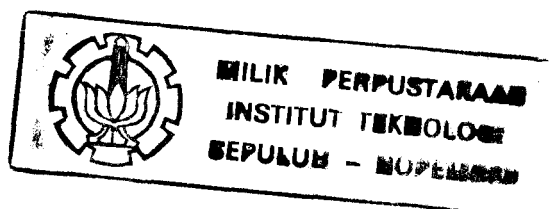
I.2. PERMASALAHAN

Anggrek yang dalam penggolongan taksonominya termasuk dalam famili *Orchidaceae*, merupakan famili yang sangat besar dan terdiri dari banyak sekali jenis atau spesies. Oleh karena itu kondisi lingkungan, cara hidup dan kebutuhan primer, yang meliputi temperatur, kelembaban, intensitas cahaya, frekuensi penyiraman, frekuensi pemupukan, untuk berbagai jenis anggrek menjadi suatu masalah yang harus diketahui.

Di Indonesia, belum adanya pembudidayaan anggrek dalam suatu rumah kaca menyebabkan perlu dilakukannya pengkajian dan studi tentang konstruksi rumah kaca yang telah ada, untuk kemudian dikembangkan suatu metode perancangan prototipe rumah kaca yang spesifik untuk budi daya anggrek. Selain itu dipikirkan adanya otomatisasi dalam pengontrolan dan pengkondisian dalam pembudidayaan sebagai pengganti cara manual.

I.3. TUJUAN

Tujuan dari pembuatan tugas akhir ini yaitu membuat suatu prototipe rumah kaca untuk budi daya anggrek, dimana didalamnya dilakukan pengontrolan dan pengkondisian temperatur, kelembaban, intensitas cahaya, frekuensi penyiraman dan pemupukan yang dapat diprogram sesuai dengan keperluan dan jenis anggrek yang dibudidayakan.



I.4. METODOLOGI

Untuk mencapai tujuan tugas akhir ini, telah dilakukan metode pendekatan sebagai berikut :

1. Melakukan penelitian lapangan pada beberapa perkebunan anggrek di Prigen, Jawa Timur.
2. Studi literatur untuk mendapatkan informasi mengenai cara hidup dan pemeliharaan tanaman anggrek.
3. Studi literatur dalam hal analisa cara teknis yang dapat dipakai sebagai alternatif yang paling cocok untuk mewujudkan solusi secara hardware.
4. Mewujudkan perangkat keras untuk sistem kontrol dengan ditunjang perangkat lunak untuk alur kontrolnya dengan interfacing ke komputer IBM PC-XT.
5. Melakukan uji coba peralatan.
6. Menganalisa data untuk menentukan solusi yang paling baik, selanjutnya diumpanbalik ke perbaikan sistem kontrol secara hardware maupun software.
7. Membuat laporan tugas akhir.

I.5. SISTEMATIKA

Pada tugas akhir ini, pembahasan dilakukan secara sistematis dengan perincian berikut ini :

- Bab I berisi latar belakang, permasalahan, tujuan, metodologi, sistematika dan relevansi dari tugas akhir yang dibuat.

- Bab II membahas teori penunjang yang diperlukan dalam perencanaan dan pembuatan tugas akhir.
- Bab III membahas perencanaan dan pembuatan prototipe rumah kaca untuk budi daya anggrek, perangkat keras elektronik dan perangkat lunak.
- Bab IV memberikan hasil pengujian alat yang dibuat beserta analisa statistiknya.
- Bab V memberikan kesimpulan dan saran dari tugas akhir yang telah dibuat.

I.6. RELEVANSI

Dari tugas akhir ini, diharapkan dihasilkan metode perancangan suatu prototipe rumah kaca untuk budi daya anggrek. Dan nantinya, dari metode yang dihasilkan, diharapkan pula dapat diterapkan pada aplikasi yang sesungguhnya.

BAB II

TEORI PENUNJANG

II.1. DESKRIPSI UMUM TANAMAN ANGGREK

Hal pertama yang perlu diperhatikan pada perencanaan pembuatan programmable kontroler untuk budi daya anggrek ini adalah data dan pengetahuan mengenai sifat, cara hidup dan kondisi lingkungan tanaman anggrek itu sendiri. Faktor-faktor yang menunjang hidup tanaman dengan baik di lingkungan aslinya seperti temperatur, kelembaban, intensitas cahaya dan lain-lain perlu diketahui untuk digunakan sebagai parameter acuan dalam menciptakan kondisi buatan yang sesuai.

II.1.1. MORFOLOGI DAN CARA HIDUP TANAMAN ANGGREK

Anggrek menurut pertumbuhannya dapat dibagi menjadi dua golongan yaitu monopodial dan simpodial. Anggrek monopodial, seperti jenis *Vanda*, *Ascocentrum*, *Doritis* dan *Phalaenopsis*, mempunyai sebuah batang dengan daun yang simetrik dan ujungnya tumbuh lurus tidak terbatas. Daun-daunnya yang tua pada batang sebelah bawah gugur dan setelah itu batang tampak mati. Pada umumnya, untuk memperbanyak dapat digunakan potongan-potongan batang tunggal tersebut karena dapat membentuk pucuk baru. Tangkai bunganya terbentuk dari batang di antara daun-

daunnya dan biasanya bergantian. Sedangkan anggrek simpodial merupakan anggrek yang pertumbuhan ujung batangnya terbatas. Setelah pertumbuhan batang maksimum, pertumbuhan baru dilanjutkan oleh anakan baru yang tumbuh di sampingnya. Batang-batang anggrek simpodial selalu merupakan tempat penyimpanan air yang disebut dengan umbi semu atau *pseudobulb*. Pada anggrek simpodial ini terdapat suatu penghubung yang disebut rhizoma yang dapat bertunas dan berkembang menjadi batang, daun dan sewaktu-waktu dapat berbunga. Yang termasuk anggrek simpodial adalah jenis *Cattleya*, *Dendrobium*, *Oncidium* dan *Paphiopedilum*.

Berdasar cara hidupnya anggrek dibedakan atas anggrek epifit dan anggrek tanah. Disebut anggrek tanah karena hanya dapat hidup di tanah atau media buatan yang diletakkan di tanah pada tempat yang terbuka. Contoh dari anggrek tanah ini misalnya *Vanda*, *Arachnis*, *Aranda* dan *Aranthera*. Sedangkan anggrek epifit, hidup menumpang pada tanaman lain tanpa merugikan tanaman yang ditumpanginya.

Di alam bebas kebanyakan anggrek bersifat epifit. Karena sifatnya itu, pembudidayaan anggrek kebanyakan berupa tanaman pot. Media tumbuhnya dapat berupa blok pakis atau pada cabang-cabang kayu. Di Indonesia, media tumbuh yang sering dipakai adalah arang, pakis, batubata dan sabut kelapa. Beberapa jenis anggrek yang biasa ditanam dalam pot adalah *Cattleya*, *Dendrobium*, *Oncidium*,

Vanda dan *Phalaenopsis*.

Peletakan untuk tanaman anggrek pot ini biasanya dengan cara digantung. Cara ini dapat memberikan keuntungan karena aliran udara akan lebih lancar. Dengan demikian CO₂ yang diperlukan dan O₂ yang dihasilkan dapat bersikulasi dengan baik. Disamping itu karena adanya aliran udara, evapotranspirasi, yaitu penguapan dari permukaan tanaman dan media, akan lebih baik. Akibatnya pengambilan air dan zat hara juga lebih baik.

II.1.2. KEBUTUHAN PRIMER TANAMAN ANGGREK

Bahwasanya famili anggrek sangatlah besar, maka cukup sulit untuk mengetahui secara mendalam tentang setiap jenis anggrek. Karenanya penelitian yang intensif terutama dalam budidaya anggrek, difokuskan pada beberapa jenis anggrek yang dewasa ini mempunyai nilai komersil yang tinggi. Penelitian menghasilkan bahwa secara garis besarnya pemupukan, penyiraman, pengendalian hama dan penyakit merupakan cara pemeliharaan yang sangat penting dilakukan. Dan agar tanaman anggrek dapat tumbuh dan berbunga memuaskan, perlu juga diatur cahaya, suhu dan kelembabannya. Berikut ini akan dijelaskan sedikit mengenai kebutuhan primer tanaman anggrek secara garis besarnya.

II.1.2.1. TEMPERATUR DAN KELEMBABAN

Kebanyakan anggrek akan tumbuh dengan baik pada temperatur yang sesuai. Pada umumnya anggrek budidaya memerlukan temperatur $28 \pm 2^{\circ}\text{C}$ dengan temperatur minim 15°C . Ada juga beberapa spesies anggrek alam yang tumbuh pada daerah pegunungan, hidup dan berkembang pada temperatur rendah yaitu $5-10^{\circ}\text{C}$.

Anggrek tanah pada umumnya lebih tahan panas dibanding anggrek pot. Tetapi temperatur yang tinggi dapat menyebabkan dehidrasi yang dapat menghambat pertumbuhan tanaman. Apabila temperatur sama atau lebih tinggi dari 30°C , maka kenaikan temperatur harus diimbangi dengan kelembaban yang tinggi serta aliran udara yang lancar.

II.1.2.2. CAHAYA

Cahaya matahari atau cahaya buatan lainnya merupakan suatu energi yang diperlukan untuk mengubah zat hara menjadi bahan yang dibutuhkan tanaman untuk tumbuh dan berkembang. Dengan demikian berarti semua tanaman membutuhkan cahaya, perbedaannya terletak pada intensitas cahaya yang dibutuhkan, panjang penyinaran serta jenis cahaya atau kualitas cahaya yang diperlukan oleh tanaman. Ketiga faktor diatas merupakan faktor cahaya yang berpengaruh terhadap pertumbuhan tanaman.

Intensitas cahaya adalah banyaknya sinar yang diterima per satuan luas. Satuannya diukur dalam lux atau footcandle. Pada umumnya anggrek pot membutuhkan intensitas cahaya yang lebih kecil dibanding anggrek tanah. Beberapa jenis anggrek tanah dapat menerima cahaya matahari penuh tanpa memerlukan peneduh. Sedangkan anggrek pot memerlukan adanya peneduh karena cahaya penuh akan menghanguskan daun dan pseudobulb sehingga akan dapat menyebabkan kematian tanaman. Anggrek-anggrek pot biasanya membutuhkan intensitas cahaya antara 1000 - 2000 lux, yaitu kira-kira 15-30% cahaya matahari penuh.

Sebagai contoh, praktisnya *Cattleya*, *Vanda* daun dan *Cymbidium* membutuhkan 20-30% cahaya matahari penuh, *Phalaenopsis* sekitar 10-15% dan *Dendrobium* dapat menerima 50-60% cahaya matahari langsung.

Sebaliknya, jika tanaman kekurangan cahaya juga menyebabkan tanaman tidak dapat berkembang dengan baik. Ciri-ciri tanaman yang kekurangan cahaya adalah daun tanaman berwarna hijau tua, daun sempit dan lunak serta tanaman tampak tinggi kurus. Warna daun yang hijau tua tersebut disebabkan tanaman mengumpulkan lebih banyak klorofil dengan tujuan menaikkan hasil fotosintesa. Daun anggrek yang menerima cukup sinar matahari akan berwarna hijau muda, helaian daunnya lebih tebal dan keras. Pada tepi daun sering terlihat ekspresi warna kemerah-merahan.

Panjang penyinaran yaitu lamanya tanaman menerima cahaya dapat mempengaruhi pembungaan tanaman dewasa. Contohnya, beberapa jenis *Cattleya* dan *Phalaenopsis* akan berbunga bila ditumbuhkan pada keadaan hari pendek yaitu penyinaran kira-kira 8 jam per hari.

Sedangkan pengaruh kualitas sinar terhadap pertumbuhan dan perkembangan tanaman jarang dilakukan. Di daerah tropis seperti halnya di Indonesia, pada umumnya cahaya putih dari matahari adalah yang terbaik. Anggrek dapat juga ditumbuhkan dalam rumah dengan sumber cahaya dari lampu fluorescent. Pada beberapa jenis *Cattleya* dan *Cymbidium*, lampu fluorescent bahkan lebih baik dari sinar matahari yang alamiah karena intensitasnya tidak terlalu tinggi dan mengandung bagian sinar yang tak berbeda dengan sinar matahari.

II.1.2.3. PENYIRAMAN

Penyiraman merupakan salah satu faktor yang kritis. Pada tanaman, air berfungsi : ¹⁾

- a. Sebagai sumber unsur H dan O yang penting untuk tanaman.
- b. Sebagai pelarut persenyawaan-persenyawaan lain.
- c. Sebagai pengisi sel.

¹⁾ Dr. Ir. Livi Winata Gunawan, *Budidaya Anggrek*, Penebar Swadaya, Jakarta, 1990, hal.36.

- d. Sebagai pengatur tekanan sel atau turgor sel.
- e. Sebagai pengangkut persenyawaan-persenyawaan dalam tubuh tanaman.
- f. Sebagai pendingin, sehingga dapat mengatur temperatur daun agar tidak naik terlalu banyak pada saat hari panas.

Kebutuhan akan air sangat tergantung dari jenis tanaman, ukuran tanaman, jenis media, jenis pot, suhu udara, kelembaban dan kecepatan angin. Pemberian air yang berlebihan seringkali merugikan anggrek. Terutama di daerah lembab, air yang berlebihan merupakan penyebab utama kematian tanaman anggrek. Kelembaban yang berlebihan dengan temperatur yang tinggi merupakan keadaan yang sangat cocok untuk pertumbuhan bakteri dan cendawan. Bakteri dan cendawan ini dapat menyerang akar tanaman dan menyebabkan busuk akar sehingga tanaman tidak lagi dapat menyerap air.

Cara pemberian air yang baik adalah melalui nozzle dari suatu semprotan. Melalui nozzle dapat diatur butiran air yang halus yang tidak menghanyutkan media tumbuh atau merusak bunga atau batang. Cara ini lebih efisien daripada penyiraman dengan penuangan air secara langsung. Air dapat disemprotkan ke media, batang dan daun tanaman hingga basah. Pada keadaan udara kering, penyemprotan butiran air halus disekeliling tanaman dan ke udara dapat

mengurangi tekanan panas yang berlebihan.

Frekuensi penyiraman ditentukan oleh jenis anggrek. Untuk anggrek-anggrek monopodial yang tidak mempunyai simpanan air dan makanan, harus disirami beberapa kali sehari pada keadaan hari panas dan kering, umumnya 2-3 kali sehari.

Tanaman yang sedang aktif tumbuh, membutuhkan lebih banyak air dibanding dengan yang sudah berbunga. Tanaman yang sedang aktif tumbuh, sel-selnya sedang membelah diri. Untuk pertumbuhan dan perkembangan sel, dibutuhkan air untuk mengisi sel-sel yang baru tersebut.

Tanaman yang sudah besar tentu lebih banyak membutuhkan air, dengan demikian volume air penyiraman harus dibedakan, juga frekuensi pemberiannya. Yang erat hubungannya dengan frekuensi penyiraman, selain ukuran tanaman adalah jenis media yang digunakan. Media sabut kelapa lebih menahan air daripada media lain, dengan demikian mengurangi frekuensi pemberian air.

Di tempat yang panas dan kering, pemberian air yang sering akan mengurangi bahaya dehidrasi tanaman sehingga metabolisme dalam tanaman dapat terus berlangsung. Umumnya pemberian air atau penyiraman dilakukan 2-3 kali dalam sehari.

Pedoman dalam hal penyiraman adalah dilakukan pada waktu permukaan media kering. Penyiraman dengan butiran

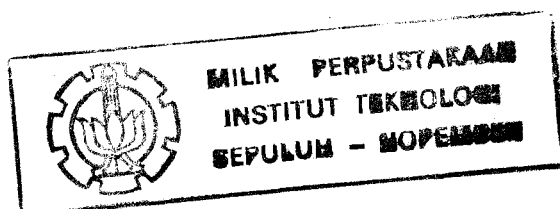
air halus dilakukan sampai membasahi seluruh pot dan menetes. Baik juga membasahi seluruh tanaman. Seringkali daun tanaman kotor dan debu-debu menutupi permukaan daun. Pencucian debu-debu ini akan menolong penyerapan sinar, air dan hara.

II.1.2.4. PEMUPUKAN

Di alam bebas, anggrek-anggrek epifit mendapatkan unsur-unsur hara dari air yang menetes atau pencucian daun-daun tanaman besar lain atau dari penguraian bahan-bahan mati dengan bantuan organisme lain. Dalam budidaya, unsur-unsur hara untuk hidup dan berkembangnya tanaman diambil dari media tumbuh dan pupuk yang diberikan. Unsur-unsur tersebut diambil melalui akar dan daun untuk kemudian diubah menjadi senyawa-senyawa organik yang dibutuhkan. Pada kultur yang menggunakan media tumbuh berupa batubata, arang atau kerikil, pemberian pupuk sangatlah penting. Frekuensi pemupukan dan jenis pupuk yang digunakan bergantung pada jenis anggrek dan keperluan.

II.1.2.5. ANGIN

Angin berfungsi sebagai penyalur karbondioksida pada siang hari dan nitrogen pada malam hari. Fungsi lainnya adalah untuk menghilangkan kantong-kantong udara



yang ada di sela-sela pot dan medium, menjaga suhu daun tetap rendah dan juga menurunkan suhu karena adanya penguapan.

II.1.2.6. PENGENDALIAN HAMA DAN PENYAKIT

Untuk mendapatkan hasil yang baik, pengendalian hama dan penyakit merupakan aspek budidaya anggrek yang tak kalah penting. Di negara tropis yang mendukung pertumbuhan tanaman sepanjang tahun juga mendukung pertumbuhan segala macam hama dan penyakit. Hama adalah hewan-hewan kecil perusak seperti serangga, tungau, ulat, keong dan lain-lain yang mengisap cairan tanaman atau memakan bagian tanaman. Sedangkan penyakit disebabkan oleh jasad renik seperti cendawan, bakteri, mikroplasma dan virus yang mengambil senyawa-senyawa dalam sel.

Suatu cara yang dapat dilakukan untuk mengurangi serangan hama dan penyakit selain dengan fungisida dan insektisida adalah dengan membuat screen house yang sederhana berupa suatu ruangan yang beratap plastik dengan dinding berupa kawat/net halus. Hama biasanya sukar dihindarkan pada pertanaman yang terbuka. Screen house ini sangat berguna untuk anggrek-anggrek yang bernilai ekonomi tinggi.

II.2. AIR CONDITIONING SYSTEM

Air Conditioning didefinisikan sebagai proses pengaturan udara dengan pengontrolan temperatur, kelembaban relatif dan distribusi udara secara terus-menerus untuk memenuhi persyaratan ruang yang dikondisikan.

Proses pengaturan itu meliputi pemanasan, penambahan uap air, pendinginan ataupun pengembunan. Perubahan yang terjadi di udara dikarenakan proses di atas dapat dilacak, dianalisa dan diprediksi dengan menggunakan *psychrometric chart*. Penggunaan *psychrometric chart* ini cukup akurat untuk tujuan praktis dan juga mudah digunakan, terutama dalam kaitannya dengan *air conditioning system*.

II.2.1. DESKRIPSI PSYCHROMETRIC CHART

Psychrometric chart ini dapat dilihat pada lampiran. Terlihat adanya hubungan antara temperatur *dry bulb* (DB), *wet bulb* (WB), kelembaban relatif dan dew point. Jika dua faktor saja telah diketahui maka dua faktor lainnya dapat dicari dari *psychrometric chart*. Selain itu, dapat juga diperoleh detail lainnya seperti entalpi, kandungan uap air, spesifik volume dan lain-lain.

Selanjutnya akan diberikan beberapa keterangan dan pengertian yang berhubungan dengan penggunaan *psychrometric chart*, diantaranya :

- **Temperatur Dry Bulb (DB)**

Skala temperatur dry bulb terletak horisontal pada bagian bawah chart. Garis vertikal dari bawah ke atas pada skala merupakan garis temperatur dry bulb yang konstan. Artinya, semua titik pada garis itu mempunyai temperatur dry bulb yang sama, sesuai dengan skala DB.

- **Temperatur Wet Bulb (WB)**

Skala temperatur wet bulb terletak sepanjang bagian luar kurva pada sisi kiri chart. Dari titik-titik pada kurva, garis wet bulb yang konstan melintas ke bawah secara diagonal ke sisi kanan dari kurva. Semua titik pada garis WB tersebut mempunyai temperatur WB yang sama.

- **Kelembaban Relatif (RH)**

Garis berupa kurva mulai dari kiri lalu menaik ke sisi kanan chart merupakan garis kelembaban relatif yang konstan. Kurva terluar pada sisi kiri adalah garis RH 100% atau kurva saturasi. Karena temperatur dry bulb, wet bulb dan dew point sama hanya ketika udara jenuh dengan uap air, maka setiap titik pada kurva saturasi ini menyatakan ketiga temperatur itu.

- **Temperatur Dew Point**

Garis dew point yang konstan melintas horisontal mulai dari kurva saturasi ke sisi kanan. Karena

temperatur wet bulb dan dew point akan sama saat saturasi (100% RH), skala WB pada kurva saturasi juga merupakan skala dew point.

- **Kandungan Uap Air**

Bagian kanan dari akhir garis dew point yang berupa garis vertikal merupakan skala spesifik humidity, yaitu banyaknya uap per pound udara kering. Karena temperatur dew point menyatakan kondisi saturasi, kandungan uap air adalah konstan pada temperatur tersebut. Oleh karena itu garis dew point juga menyatakan garis spesifik humidity.

- **Entalpi**

Entalpi atau kandungan panas total dari udara hanya bergantung pada temperatur wet bulb. Meskipun suatu sampel udara memiliki temperatur dry bulb yang berbeda tetapi temperatur wet bulb-nya sama maka kandungan panas totalnya tetap sama, hanya saja proporsi dari *sensible heat* dan *latent heat* dari panas total yang bervariasi. Pada psychrometer chart skala entalpi terletak bersama temperatur WB pada kurva saturasi. Kandungan panas total ini dinyatakan dalam BTU/lb udara kering.

- **Spesifik Volume**

Berat udara kadangkala diperlukan untuk tujuan perhitungan. Tetapi dalam aplikasinya, yang biasa

dipakai adalah volume udara. Karenanya, dalam psychrometric chart garis spesifik volume diberikan untuk mendapatkan volume udara per satuan berat. Spesifik volume atau kerapatan udara berubah tergantung pada temperatur. Garis spesifik volume dimulai dari kurva saturasi dan menurun dengan sudut yang tajam ke garis vertikal.

II.2.2. METODA PENGGUNAAN PSYCHROMETRIC CHART

Untuk pemahaman akan penggunaan psychrometric chart, akan lebih mudah jika penjelasan dinyatakan langsung dengan suatu contoh. Dimisalkan telah diketahui suhu udara saat itu yaitu DB 85°F dan WB 70°F , seperti tampak pada gambar 2.1 dan dinyatakan dengan garis kontinyu. Faktor-faktor lain seperti kelembaban relatif, entalpi, kandungan uap air, dew point dan spesifik volume selanjutnya dapat dicari pada chart tersebut yang dinyatakan dengan garis putus-putus.

Prosedurnya adalah sebagai berikut :

- Tarik garis yang menyatakan garis DB 85°F dan WB 70°F . Titik perpotongannya dinamakan titik O yaitu menyatakan kondisi temperatur 85°F dry bulb dan 70°F wet bulb.
- Terlihat titik O terletak antara kurva RH 40% dan 50%. Pendekatan diambil sehingga didapatkan kelembaban relatif udara adalah 48%.

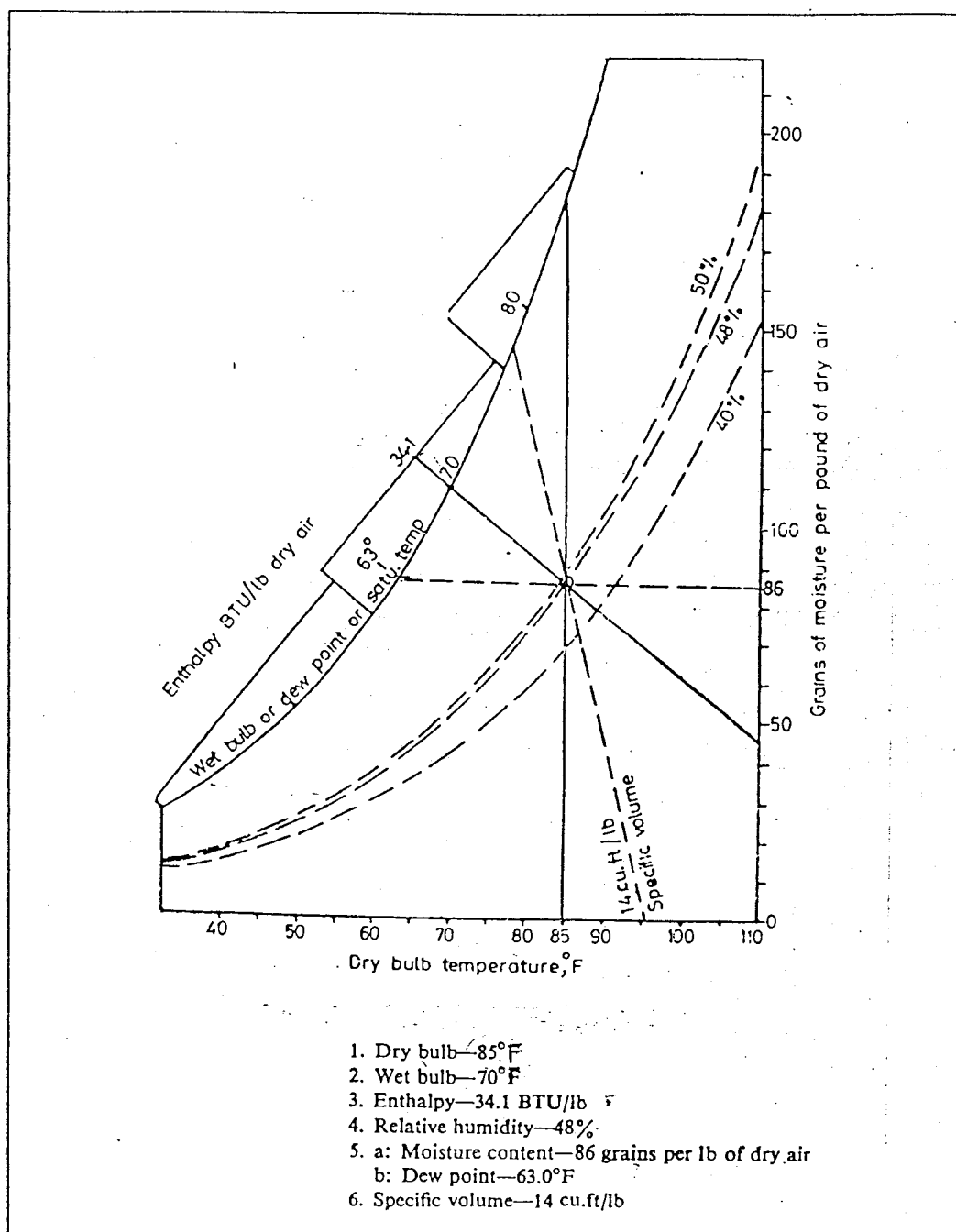
- Dengan memperpanjang garis WB 70°F ke skala entalpi, didapatkan kandungan panas total udara adalah 34.1 BTU/lb.
- Ditarik garis horisontal melalui titik O hingga mencapai kurva saturasi di kiri dan skala spesifik humidity di kanan. Dengan itu didapatkan temperatur dew point adalah 63°F dan kandungan uap air sebesar 86 grains per pound udara kering.
- Ditarik garis spesifik volume melalui titik O sehingga dengan itu didapat spesifik volume sebesar 14 cu.ft/lb.

Jadi hanya dengan mengetahui dua faktor dari enam faktor tersebut, dengan psychrometric chart akan dapat dicari faktor lainnya. Dengan demikian prediksi pada air conditioning system akan lebih mudah dilakukan.

II.2.3. HEATING DAN HUMIDIFYING

Jika udara dipanasi dan uap air ditambahkan maka proses ini dinamakan *heating* dan *humidification*. Proses ini dapat dinyatakan sebagai dua proses yang berbeda yaitu pemanasan (*heating*) dan penambahan uap air.

Dengan pemanasan berarti ditambahkan panas ke udara sehingga temperatur udara naik. Tidak ada uap air yang ditambahkan, karenanya kandungan uap tetap. Jadi panas yang ditambahkan hanya menaikkan sensible heat dari udara, dengan demikian menaikkan kandungan panas total



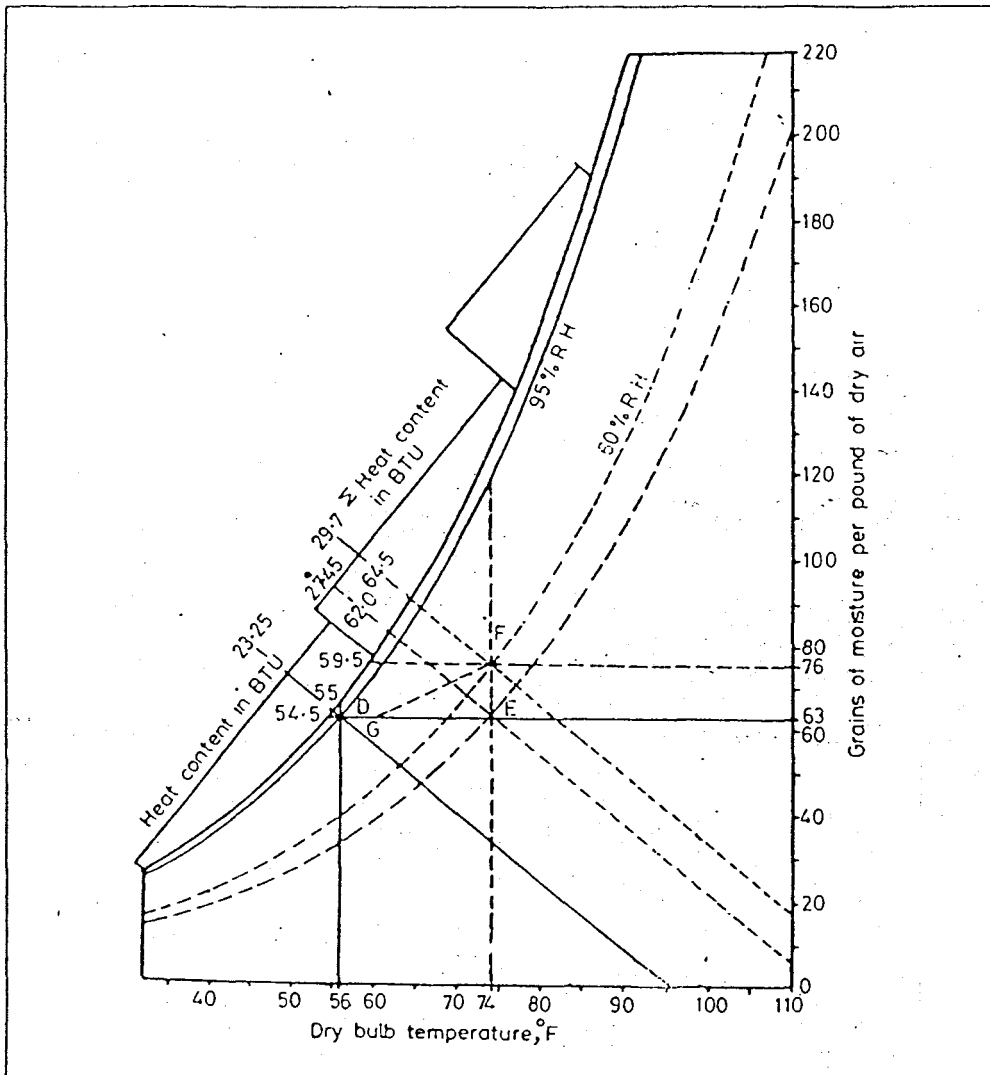
GAMBAR 2.1

METODA PENGGUNAAN PSYCHROMETRIC CHART

dari udara. Sedangkan penambahan uap air saja akan menaikkan latent heat, karenanya kandungan panas total juga naik.

Untuk mengetahui perubahan apa saja yang terjadi pada proses heating dan humidifying ini, maka pada gambar 2.2 ditunjukkan alur proses tersebut. Kondisi udara mula-mula pada 56°F DB dan 55°F WB, kemudian dipanasi sampai mencapai 74°F DB. Karena tidak ada penambahan uap air maka kandungan uap airnya tetap yaitu 63 grains/lb, demikian juga halnya dengan temperatur dew point tetap pada 54.5°F . Jadi proses heating dapat dinyatakan dengan garis horisontal dimulai dari titik D dan memotong garis vertikal 74°F DB pada titik E. Dengan menarik garis melintasi titik E ini, temperatur wet bulb dapat dibaca yaitu 62°F WB. Kandungan panas total mula-mula adalah 23.25 BTU/lb dan setelah pemanasan dapat dibaca sebesar 27.45 BTU/lb. Jadi ada penambahan panas sebesar $27.45 - 23.25 = 4.2$ BTU/lb. Adapun besarnya kelembaban relatif setelah pemanasan turun dari 95% menjadi 50%.

Kemudian pada kondisi akhir pemanasan, dilakukan proses humidifying yaitu dengan menginjeksikan uap air ke sistem sehingga ada penambahan uap air dari 63 grains/lb menjadi 76 grains/lb. Penambahan uap air ini tidak mengubah besar sensible heat sehingga dry bulb konstan pada 74°F DB.



GAMBAR 2.2

PROSES HEATING DAN HUMIDIFYING

Jadi proses ini dapat dinyatakan dengan garis vertikal dan didapat titik F sebagai hasil perpotongannya.

Dari titik ini didapatkan temperatur WB naik dari 54.5°F menjadi 59.5°F, kandungan panas total naik sebesar 2.25

BTU/lb dari 27.45 BTU/lb menjadi 29.7 BTU/lb, kelembaban relatif naik dari 50% menjadi 60%. Adapun hasil proses heating dan humidifying di atas dirangkum sebagaimana dapat dilihat pada tabel 2-1.

Jadi pada gambar 2.2, garis DE menyatakan proses heating, garis EF menyatakan proses humidification dan garis DF akan menyatakan kombinasi dari pengaruh proses heating dan humidification.

TABEL 2-1

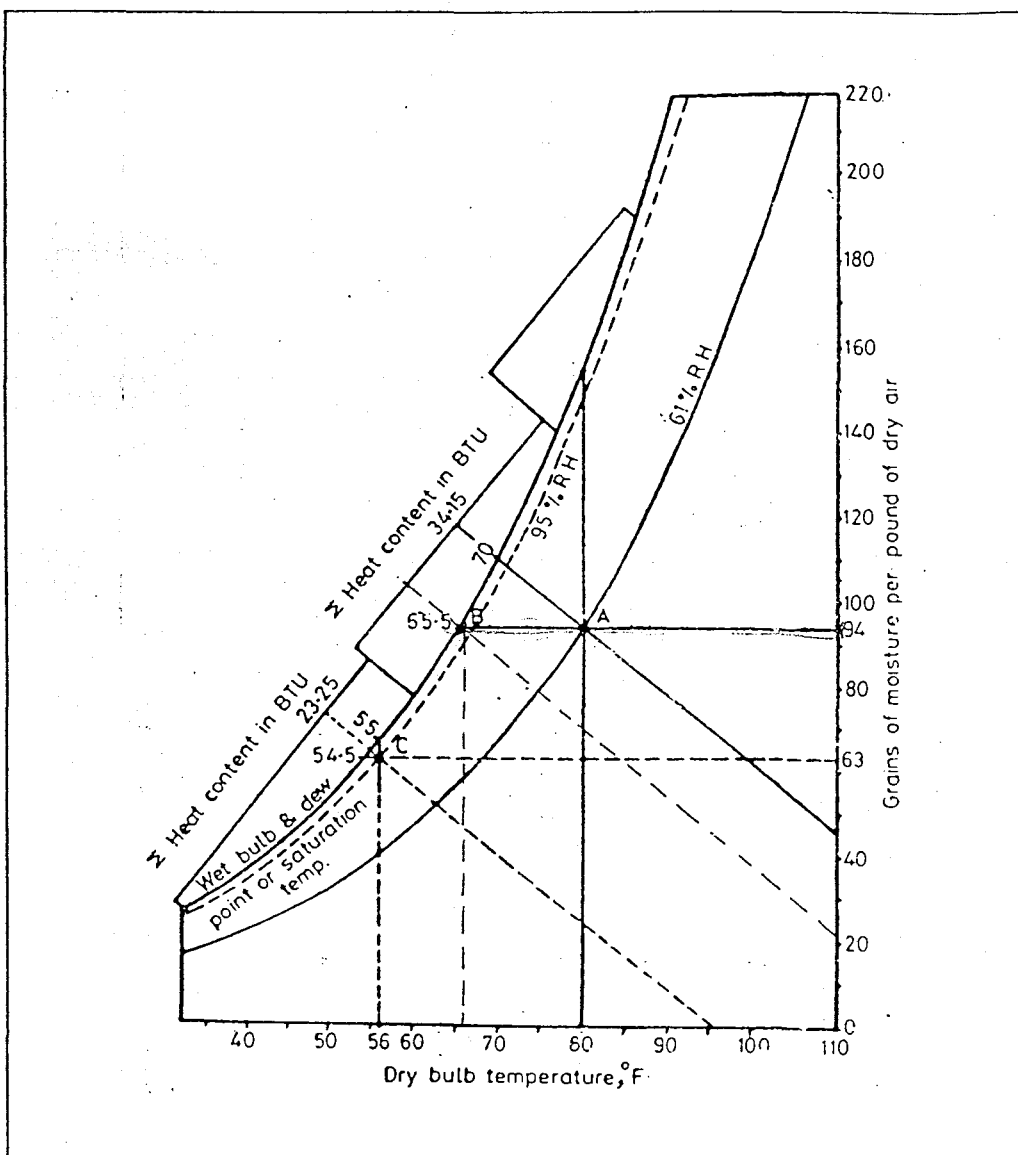
HASIL PROSES HEATING DAN HUMIDIFYING

	Kondisi mula-mula	Kondisi setelah pemanasan	Kondisi akhir
Dry bulb (°F)	56	74	74
Wet bulb (°F)	55	62	64.5
RH (%)	95	50	60
Dew point (°F)	54.5	54.5	59.5
Entalpi (BTU/lb)	23.25	27.45	29.7
Kandungan uap air (grains/lb)	63	63	76

II.2.4. COOLING DAN DEHUMIDIFICATION

Proses *cooling* mengacu pada sensible cooling yaitu tidak adanya pengembalian latent heat. Pelepasan latent heat dari udara terjadi ketika uap air yang terkandung dalam udara dikondensasi sehingga mengurangi jumlah uap air di udara. Hal ini hanya dimungkinkan ketika udara didinginkan di bawah temperatur dew pointnya. Jadi bila dikatakan bahwa udara hanya didinginkan tanpa ada

pelepasan latent heat maka udara didinginkan hanya pada temperatur di atas temperatur dew pointnya. Sedangkan dehumidification disertai dengann pelepasan latent heat, maka udara didinginkan di bawah temperatur dew pointnya.

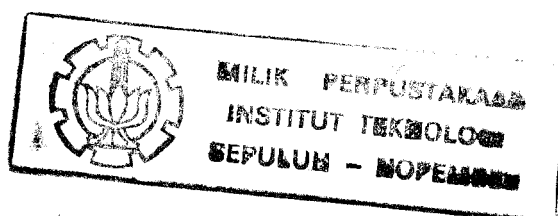


Gambar 2.3 menyatakan proses cooling dan dehumidification pada chart untuk udara dari 80°F DB dan 70°F WB ke 56°F DB. Untuk memudahkan pengertian, proses ini dapat dibedakan menjadi dua proses terpisah yaitu pertama, cooling dari 80°F DB dan 70°F WB ke temperatur dew point yaitu 65.5°F. Proses ini merupakan sensible cooling dan dinyatakan oleh garis AB yang berupa garis horisontal (dew point dan kandungan uap air tetap). Proses kedua merupakan cooling (dan dehumidifying) dari 65.5°F ke 56°F DB dan dinyatakan dengan kurva BC. Garis AC menyatakan proses yang lengkap. Adapun kondisi dari udara sebelum dan sesudah proses ditabelkan pada tabel 2-2.

TABEL 2-2

HASIL PROSES COOLING DAN DEHUMIDIFICATION

	Kondisi mula-mula	Kondisi setelah cooling	Kondisi akhir
Dry bulb (°F)	80	65.5	56
Wet bulb (°F)	70	65.5	55
RH (%)	61	100	95
Dew point (°F)	65.5	65.5	54.5
Entalpi (BTU/lb)	34.15	30.5	23.25
Kandungan uap air (grains/lb)	94	94	63



II.3. PERPINDAHAN KALOR

Ada tiga mekanisme dasar untuk perpindahan tenaga sebagai kalor yaitu konduksi (hantaran), konveksi (aliran) dan radiasi (pancaran). Konduksi merupakan suatu proses perpindahan tenaga sebagai kalor melalui sebuah medium dimana bagian zat tidak berpindah (*stationer*), seperti tembaga, air atau udara. Bila bagian zat ikut berpindah maka proses perpindahan kalor itu dikenal dengan proses konveksi. Sedangkan radiasi merupakan suatu cara perpindahan kalor tanpa memerlukan medium, dimana kalor dipancarkan dalam bentuk gelombang elektromagnetik.

II.3.1. KONDUKSI

Pada permasalahan praktis yang umum ditemui, panas mengalir dari suatu medium melalui suatu bahan padat menahan panas dan keluar ke medium lainnya. Di sini terjadi suatu proses perpindahan kalor yang disebut dengan proses konduksi.

Tinjauan difokuskan pada *steady conduction* yaitu perpindahan kalor dalam keadaan *steady state* dimana temperatur pada suatu titik tidak bergantung pada waktu. Oleh Fourier, mekanisme perpindahan kalor dengan *steady conduction* dinyatakan dalam hubungan berikut :

$$q = -k A \frac{dt}{dx} \quad \dots (2-1)$$

dimana q adalah kecepatan konduksi sepanjang sumbu x , A adalah penampang melintang dari bidang yang tegak lurus sumbu x , $-dt/dx$ adalah gradien temperatur sepanjang bidang dan k adalah konduktivitas termal yang merupakan sifat fisik dari zat. Pada permasalahan praktis, luas penampang melintang A bervariasi dengan ketebalan bidang x , sehingga bentuk dari bidang yang dilalui oleh aliran panas haruslah diketahui terlebih dahulu.

Untuk konduksi kalor melalui suatu bahan padat dengan penampang melintang A yang tetap, persamaan kecepatan konduksinya adalah sebagai berikut : 2)

$$q = \frac{k_m A \Delta t}{x} \quad \dots (2-2)$$

dimana :

- q = kecepatan perambatan kalor konduksi (Btu/h)
- k_m = konduktivitas termal medium (Btu/h.ft.°F)
- A = luas penampang (ft²)
- $\Delta t/x$ = gradien suhu rata-rata (°F/ft)

Persamaan 2-6 menunjukkan bahwa kecepatan rambatan kalor secara konduksi berbanding lurus dengan konduktivitas termal rata-rata medium, luas penampang bidang, beda temperatur antara titik-titik dari dan kemana kalor merambat serta berbanding terbalik dengan panjang atau ketebalan bidang.

2) McAdams, William H, *Heat Transmission*, McGraw-Hill, Singapore, 1985, hal.12

II.3.2. KONVEKSI

Terkait dengan air conditioning system dalam suatu ruang yang dikondisikan, maka proses konveksi menjadi suatu faktor yang penting. Bila fluida lewat di atas sebuah permukaan padat panas maka tenaga dipindahkan kepada fluida dari dinding oleh proses hantaran. Tenaga ini kemudian dikonveksikan dan didifusikan melalui fluida oleh hantaran di dalam fluida tersebut.

Jika proses aliran fluida itu diinduksi oleh sebuah pompa atau sistem pengedar lain (circulating system), maka digunakan istilah konveksi yang dipaksakan atau *forced convection*. Sebaliknya bila aliran fluida timbul karena daya apung fluida yang disebabkan oleh pemanasan, maka proses tersebut dinamakan dengan konveksi bebas atau *natural convection*.

Pada proses konveksi, jumlah kalor yang dialirkan dalam medium, baik berupa zat cair atau gas, tergantung pada koefisien konveksi, penampang yang dilalui dan beda suhunya. Besarnya kalor yang dipindahkan ini dinyatakan dengan rumus :

$$q = h_m \cdot A \cdot \Delta T \quad \dots (2-3)$$

dimana :

- q = kalor yang dipindahkan (Btu/h)
- h_m = koefisien perpindahan kalor konveksi (Btu/h.ft².°F)
- A = luas penampang (ft²)
- ΔT = beda temperatur (°F)

Besarnya koefisien konveksi dibedakan oleh dimensi permukaan sebagai karakteristik faktor geometris. Untuk udara pada temperatur ruang dan tekanan atmosfer, koefisien dapat dicari berdasar rujukan berikut :

a. Vertical Plate

$$10^3 > L^3(\Delta t) > 10^{-2} \quad \rightarrow \quad h_m = 0.28 [(\Delta t)/L]^{1/4} \quad \dots (2-4)$$

$$L^3(\Delta t) > 10^2 \quad \rightarrow \quad h_m = 0.19 (\Delta t)^{1/3} \quad \dots (2-5)$$

b. Horizontal Cylinder

$$10^2 > D^3(\Delta t) > 10^{-3} \quad \rightarrow \quad h_m = 0.25 [(\Delta t)/D]^{1/4} \quad \dots (2-6)$$

$$D^3(\Delta t) > 10^2 \quad \rightarrow \quad h_m = 0.19 (\Delta t)^{1/3} \quad \dots (2-7)$$

c. Heated Horizontal Plate Facing Upward atau Cooled Horizontal Plate Facing Downward

$$10 > L^3(\Delta t) > 0.1 \quad \rightarrow \quad h_m = 0.27 [(\Delta t)/L]^{1/4} \quad \dots (2-8)$$

$$10^4 > L^3(\Delta t) > 10 \quad \rightarrow \quad h_m = 0.22 (\Delta t)^{1/3} \quad \dots (2-9)$$

d. Heated Horizontal Plate Facing Downward atau Cooled Horizontal Plate Facing Upward

$$10^4 > L^3(\Delta t) > 0.1 \quad \rightarrow \quad h_m = 0.12 [(\Delta t)/L]^{1/4} \quad \dots (2-10)$$

dimana :

h_m = koefisien konveksi (BTU/h.ft².°F)
 D = diameter silinder (ft)

Adapun L (dalam satuan feet, ft) dicari dari hubungan :

$$\frac{1}{L} = \frac{1}{L_{\text{vert}}} + \frac{1}{L_{\text{horis}}} \quad \dots (2-11)$$

L_{vert} dan L_{horis} merupakan panjang sisi vertikal dan panjang sisi horisontal dari bidang.

II.4. TRANSDUSER

Pengontrolan dan pengaturan dari sistem industri dan proses-proses industri bergantung pada keakuratan pengukuran. Karenanya pengontrolan terhadap variabel yang diukur harus akurat. Dalam sistem industri, peralatan yang melakukan pengukuran ini disebut dengan transduser atau sensor. Dapat pula dinyatakan bahwa transduser adalah peralatan yang mengubah besaran yang diukur menjadi keluaran yang menunjang pengukuran. Dengan kata lain, transduser mengubah suatu variabel (aliran fluida, temperatur, kelembaban) menjadi variabel yang analog.

II.4.1. SENSOR TEMPERATUR

Temperatur dapat diukur dengan banyak cara. Untuk memudahkan, sensor temperatur dikelompokkan menjadi 2 bagian yaitu sensor mekanis dan sensor elektris.

Sensor temperatur mekanis merupakan peralatan yang mengubah temperatur menjadi suatu gerakan atau posisi. Sensor jenis ini bergantung pada prinsip fisika bahwa gas, cairan dan padatan berubah volumenya ketika dipanaskan. Untuk zat yang berbeda, berbeda pula jumlah perubahan volumenya. Contoh dari jenis sensor mekanis ini yaitu Glass-Stem Thermometer, Filled-System Thermometer, dan Bimetallic Thermometer.

Untuk penggunaan dalam sistem kontrol proses analog,

sensor mekanis ini tidak cocok karenanya digunakan sensor temperatur elektris. Yang termasuk dalam jenis ini adalah thermocouple, thermistor, resistance temperature detector (RTD) dan sensor semikonduktor atau IC sensor. Masing-masing sensor ini memiliki kekurangan dan kelebihan seperti yang ditunjukkan pada tabel 2-3.

SENSOR IC TEMPERATUR





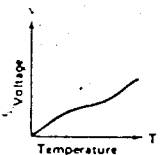
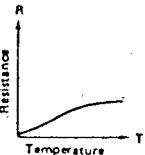
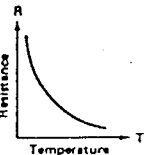
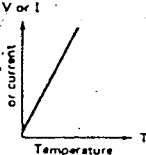
Keluaran dari sensor IC temperatur proporsional terhadap temperatur. IC sensor menghasilkan linearitas yang sangat baik, kecil bentuknya, murah harganya serta menghasilkan keluaran yang sangat sensitif. IC sensor ini mampu mengeliminasi kesalahan linear yang merupakan kelemahan dari sensor thermocouple dan thermistor. Namun kerugian dari IC sensor ini adalah dapat terjadinya kesalahan akibat self-heating. Adapun cara mengurangi efek ini adalah dengan mengoperasikannya pada arus minimum namun cukup untuk mendrive sensor.

Pada dasarnya IC sensor ini ada 2 jenis yaitu sumber tegangan sensitif terhadap temperatur dan sumber arus sensitif terhadap temperatur. Pada jenis IC sensor sumber tegangan sensitif terhadap temperatur, tegangan berubah terhadap perubahan temperatur. Contoh dari jenis ini diantaranya LM34, LM35, LM135, LM235 dan LM335. Sedang untuk jenis sumber arus, arus berubah terhadap perubahan

temperatur dan salah satu contohnya adalah LM334. Adapun keuntungan dari sensor sumber arus ini yaitu tidak terpengaruhnya keluaran sensor akibat adanya penurunan tegangan dalam jalur transmisi yang panjang.

TABEL 2-3 3)

PERBANDINGAN SENSOR TEMPERATUR ELEKTRIS

	Thermocouple	RTD	Thermistor	IC Sensor
				
				
Advantages	<input type="checkbox"/> Self-powered <input type="checkbox"/> Simple <input type="checkbox"/> Rugged <input type="checkbox"/> Inexpensive <input type="checkbox"/> Wide variety <input type="checkbox"/> Wide temperature range	<input type="checkbox"/> Most stable <input type="checkbox"/> Most accurate <input type="checkbox"/> More linear than thermocouple	<input type="checkbox"/> High output <input type="checkbox"/> Fast <input type="checkbox"/> Two-wire ohms measurement	<input type="checkbox"/> Most linear <input type="checkbox"/> Highest output <input type="checkbox"/> Inexpensive
Disadvantages	<input type="checkbox"/> Non-linear <input type="checkbox"/> Low voltage <input type="checkbox"/> Reference required <input type="checkbox"/> Least stable <input type="checkbox"/> Least sensitive	<input type="checkbox"/> Expensive <input type="checkbox"/> Power supply required <input type="checkbox"/> Small ΔR <input type="checkbox"/> Low absolute resistance <input type="checkbox"/> Self-heating	<input type="checkbox"/> Non-linear <input type="checkbox"/> Limited temperature range <input type="checkbox"/> Fragile <input type="checkbox"/> Power supply required <input type="checkbox"/> Self-heating	<input type="checkbox"/> $T < 200^\circ\text{C}$ <input type="checkbox"/> Power supply required <input type="checkbox"/> Slow <input type="checkbox"/> Self-heating <input type="checkbox"/> Limited configurations

LM335 merupakan jenis yang mudah didapat di pasaran dan harganya cukup murah. Sensor ini memiliki range pengukuran dari -25°C sampai 100°C . LM335 ini bekerja sebagai zener dua terminal dengan output sebanding dengan

3) Charles A. Schuler, William L. McNamee, *Industrial Electronics and Robotics*, 1-st edition, McGraw-Hill, Singapore, 1986, hal.214.

II.4.2. SENSOR KELEMBABAN RELATIF

Banyak proses industri bergantung pada keakuratan pengontrolan kelembaban. Kelembaban didefinisikan sebagai jumlah kandungan uap air di udara. Seberapa banyak uap air yang dikandung bergantung pada beberapa faktor. Faktor terpenting adalah temperatur udara. Udara hangat mengandung uap air lebih banyak daripada udara dingin.

Pengukuran kandungan uap air di udara umumnya memakai apa yang disebut kelembaban relatif. Kelembaban relatif didefinisikan sebagai perbandingan antara jumlah uap air yang dikandung pada temperatur tertentu dengan kandungan uap air maksimum saat itu.

Sensor yang sering dipakai untuk mengukur kelembaban relatif yaitu psychrometer, hygrometer serat, hygrometer impedansi termasuk hygrometer resistansi dan resistansi-kapasitansi, hygrometer redaman frekuensi dan hygrometer kapasitif. Tabel 2-4 menunjukkan perbandingan sensor kelembaban tersebut.

Untuk aplikasi praktis, penggunaan metode psychrometer untuk mengukur kelembaban relatif memberikan hasil yang cukup akurat. Konstruksi psychrometer ini terdiri atas dua bulb, satu kering (dry bulb) dan satu basah (wet bulb). Wet bulb dikondisikan dengan membungkus sensor dengan sumbu basah. Udara harus dilewatkan melalui wet bulb yang menyebabkan air menguap.

Kedua bulb tersebut mengandung sensor temperatur. Jika temperatur udara tetap konstan, temperatur dry bulb sensor tetap. Tetapi wet bulb bervariasi menurut kelembaban relatif. Makin cepat kecepatan penguapan yang terjadi, efek pendinginan pada wet bulb makin cepat sehingga pembacaan temperatur pada wet bulb makin kecil. Kecepatan penguapan bergantung pada kelembaban relatif dari udara yang mengalir. Jika udara lembab, kecepatan penguapan tidaklah tinggi karenanya wet bulb hanya sedikit lebih dingin dari dry bulb. Perbedaan temperatur dari dry bulb dan wet bulb inilah yang merefleksikan harga kelembaban relatif dari udara.

TABEL 2-4 5)

PERBANDINGAN SENSOR KELEMBABAN

<i>Name</i>	<i>Operating Principle</i>	<i>Relative Humidity Range</i>	<i>Temperature Range</i>	<i>Accuracy</i>
Psychrometer	Evaporation rate of water	2%-98%	32°-212°F	±5%
Hair hygrometer	Hair changes length with humidity	15%-90%	0°-160°F	±5%
Resistance hygrometer	Lithium chloride changes resistance with humidity	1.5%-99%	-40°-160°F	±1.5%
Resistance-capacitance hygrometer	Aluminum oxide changes impedance with humidity	0%-100%	0°-300°F	±2%
Capacitive hygrometer	Dielectric constant changes with humidity	0%-100%	-200°-200°F	±0.5%

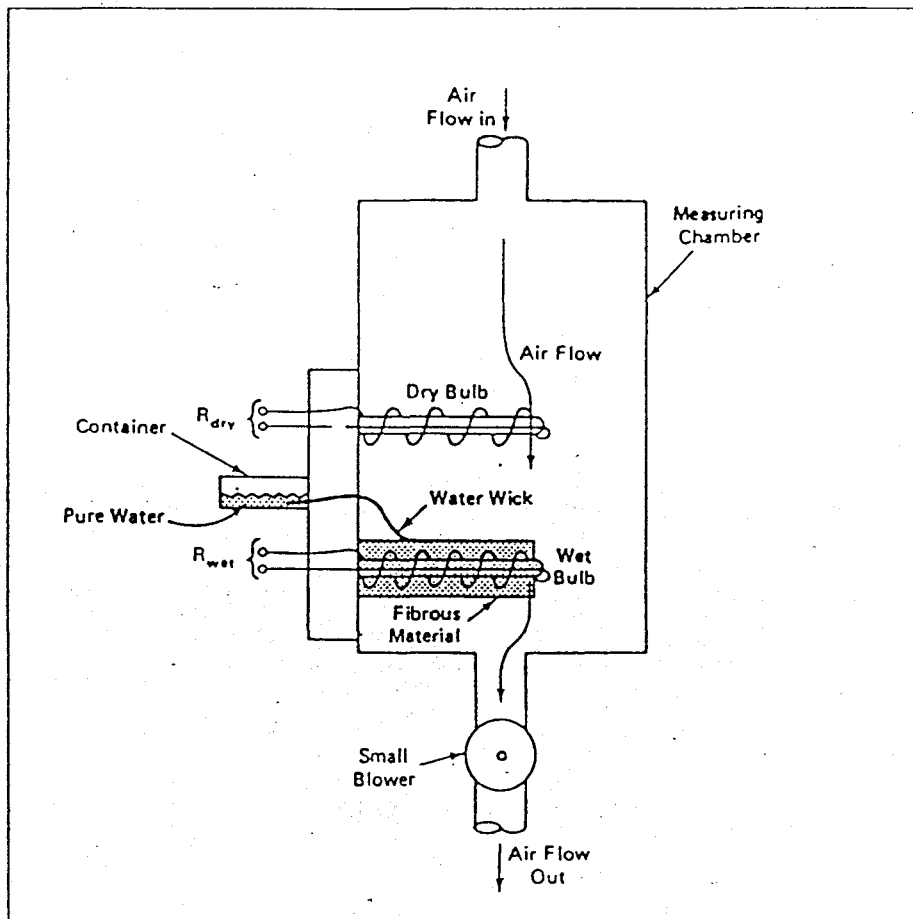
5) James T. Humpries, Leslie P. Sheets, *Industrial Electronics*, Breton Publishers, Massachusetts, USA, 1983, hal.162.

Gambar 2.5 menunjukkan pemodelan pengukuran kelembaban dengan psychrometer. Untuk keakuratan penggunaan metoda ini, pembacaan temperatur pada wet bulb harus benar. Oleh karena itu, pada perencanaan dan konstruksinya, kondisi yang harus dipenuhi yaitu : ⁶⁾

1. Sumbu harus dijaga agar tetap basah dan mengandung air murni.
2. Air harus mencapai wet bulb pada temperatur wet bulb, tidak lebih hangat atau lebih dingin.
3. Kecepatan udara melintasi sumbu harus cukup, yaitu sekitar 600 feet per menit atau lebih tinggi.
4. Tidak terjadi konduksi panas ke atau dari bagian sensitif wet bulb dengan tempat peletakannya.
5. Tidak terjadi perpindahan panas ke atau dari wet bulb yang disebabkan oleh radiasi.

Jika hasil pembacaan ingin dikonversikan ke besaran kelembaban relatif maka ada dua kondisi lagi yang harus dipenuhi yaitu pembacaan dry bulb yang akurat dan tidak dipengaruhi oleh pendinginan wet bulb serta adanya tabel pembacaan atau chart yang benar. Interpretasi penggunaan chart atau tabel inilah yang merupakan keterbatasan dari psychrometer ini. Chart yang digunakan disebut dengan psychrometric chart sebagaimana tampak pada lampiran A-1.

⁶⁾ Considine, Douglas M., ed, *Process Instruments and Controls handbook*, second edition, McGraw-Hill, Inc., USA, 1974.



GAMBAR 2.5 7)

PENGUKURAN KELEMBABAN RELATIF DENGAN PSYCHROMETER

II.4.3. SENSOR CAHAYA

Sekarang ini sistem dan peralatan optoelektronik telah menjadi bagian integral dari sistem kontrol di industri. Kepopuleran peralatan optoelektronik didasarkan

7) Timothy J. Maloney, *Industrial Solid-state Electronics*, Second Edition, Prentice-Hall, Inc., New Jersey, USA, 1986, hal.451.

atas kenyataan bahwa peralatan tersebut merupakan sensor yang tidak kontak secara langsung. (non-contact sensors).

Sensor optoelektronik ini dapat dibagi menjadi tiga bagian yaitu sensor photoconductive, sensor photovoltaic dan sensor photoemissive. Sensor photoconductive merupakan sensor yang mengubah perubahan intensitas cahaya menjadi konduktivitas. Contohnya adalah photoresistor, photodiode, phototransistor dan photodarlington. Sedangkan sensor photovoltaic merupakan sensor yang menghasilkan tegangan ketika disinari cahaya. Dan terakhir, sensor photoconductive akan melepaskan elektron ketika disinari. Penggunaannya biasanya pada tabung hampa dan photosensor seperti phototube dan photomultiplier. Perbandingan dari ketiga jenis sensor optoelektronik ini ditabelkan pada tabel 2-5.

Terdapat dua tipe dari sensor photovoltaic yaitu silicon cell dan selenium cell. Pada intensitas cahaya yang tinggi, silicon cell menghasilkan maksimum 0,5 V pada kedua terminalnya dan mensuplai arus maksimum 100 mA. Selenium cell meskipun memiliki respon spektrum yang lebih baik, kurang efisien sebagai pengubah daya. Karena itu silicon cell lebih berguna sebagai solar cell. Kegunaan dari solar cell adalah mengubah energi radiasi matahari menjadi energi listrik. Solar cell sering dipasang secara seri, paralel ataupun kombinasi seri-

paralel untuk memperbesar arus dan tegangan keluaran.

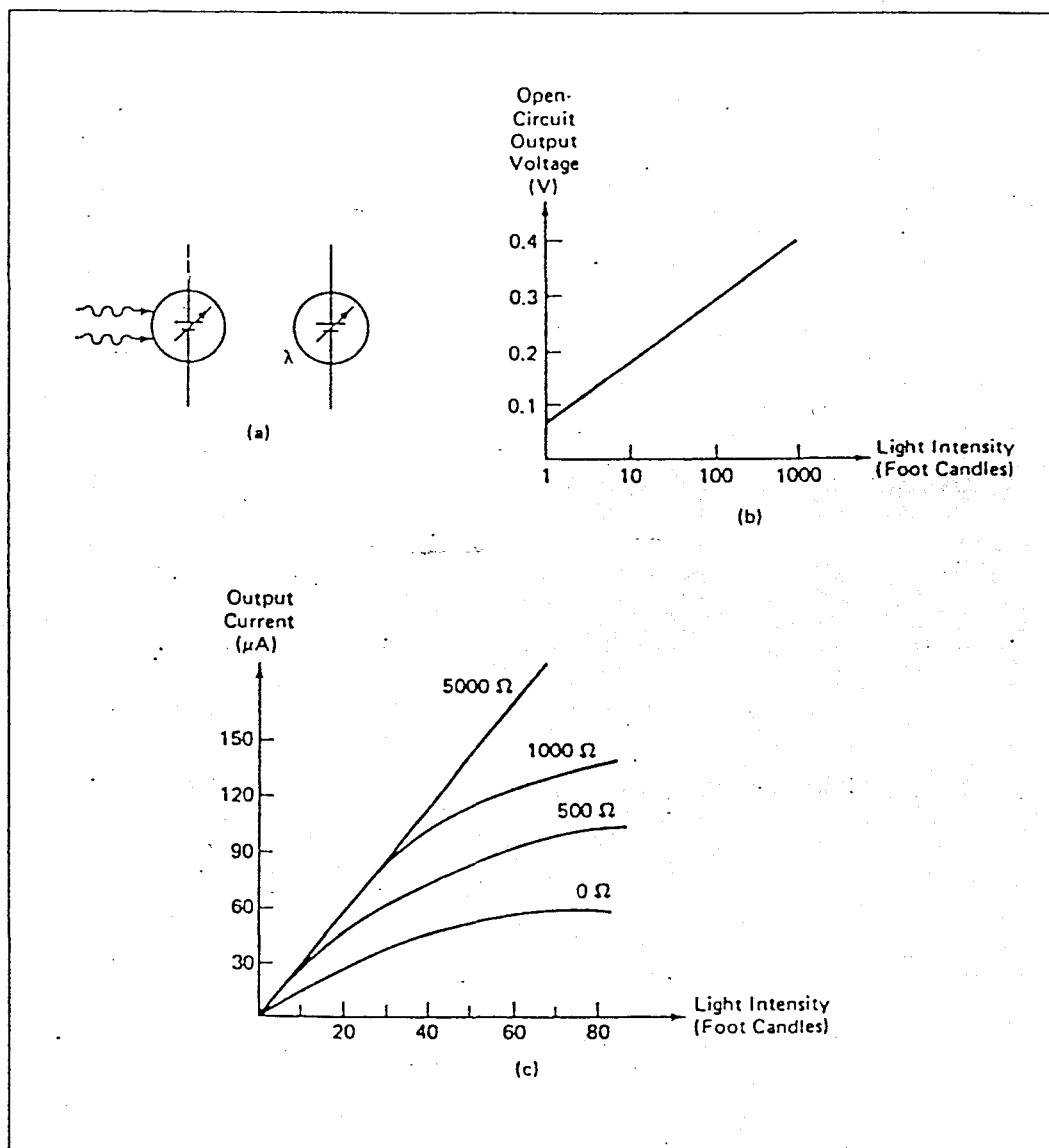
Pada gambar 2.6 ditunjukkan simbol yang biasa digunakan untuk photovoltaic (solar) cell dan karakteristik dari photovoltaic cell yang meliputi tegangan keluaran open circuit vs intensitas cahaya dan arus keluaran vs intensitas cahaya untuk beban tertentu.

TABEL 2-5 B)

PERBANDINGAN SENSOR CAHAYA

Type	Name	Spectral Response (μm)	Response Time	Advantages	Disadvantages
Photoconductive	Photoresistor	0.5–0.7	100 ms	Small	Slow
	CdS	0.5–0.7	100 ms	High sensitivity	Hysteresis
	CdSe	0.6–0.9	10 ms	Low cost	Temperature range limited
	Photodiode	0.4–0.9	1 ns	Visual range	
				Very fast	Low-level output (current)
				Good linearity	
				Low noise	
				Wide spectral range	
	Phototransistor	0.25–1.1	1 μs	High current gain	Low frequency response (500 KHz)
				Can drive TTL	Nonlinear
				Small size	
Photovoltaic	Photovoltaic (solar) cell	0.35–0.75	20 μs	Linear	Slow
				Self-powered	Low-level output (voltage)
				Visual range	
Photoemissive	Phototube	0.15–1.2	1 μs	Good stability	Fragile
				Fast response	
				High impedance	
	Photomultiplier	0.2–1.0	1–10 ns	Fast response	Fragile
				High sensitivity	Need high-voltage power supply
				Good linearity	
				Wide spectral range	

B) James T. Humpries, Leslie P. Sheets, op.cit, hal.175.

GAMBAR 2.6 ⁹⁾

NOTASI DAN KARAKTERISTIK PHOTOVOLTAIC CELL

⁹⁾ Timothy J. Maloney. op.cit, hal.419.

II.5. KONTROLER DIGITAL

Terdapat dua sistem kontrol yang dipakai di dalam kontroler untuk mengontrol suatu proses yaitu kontrol loop terbuka dan kontrol loop tertutup. Kontrol loop tertutup lebih lazim digunakan karena lebih teliti dalam sistem pengontrolan karena adanya umpan balik dari keluaran ke masukan.

Kontroler yang konvensional sering disebut dengan kontrol analog, baik yang menggunakan penguat operasional sampai yang menggunakan komputer analog. Dewasa ini kebutuhan akan kontrol memerlukan fasilitas yang lebih baik, antara lain seperti dapat memproses masukan yang lebih banyak, waktu pemrosesan yang cepat dan lain-lain. Hal tersebut sulit dicapai oleh kontroler analog, karena itu dikembangkan kontroler digital.

Dalam industri terdapat beberapa klasifikasi pengontrolan yaitu :

1. Kontrol dua posisi / ON-OFF.
2. Kontrol Proporsional.
3. Kontrol Integral.
4. Kontrol Proporsional plus integral (PI)
5. Kontrol Proporsional plus Derivatif (PD)
6. Kontrol Proporsional plus Integral plus Derivatif (PID).

II.5.1. KONTROL DUA POSISI / ON-OFF

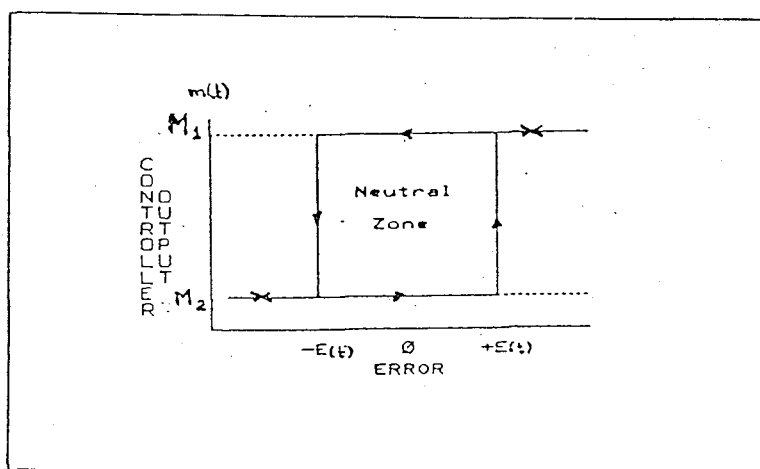
Dalam kontrol dua posisi, elemen penggerak hanya mempunyai dua posisi tetap, yaitu "on" dan "off". Kontrol dua posisi ini relatif sederhana dan murah, karenanya banyak digunakan dalam sistem kontrol.

Untuk sinyal keluaran $m(t)$, sinyal kesalahan penggerak adalah $e(t)$. Pada jenis kontrol ini, $m(t)$ akan tetap pada harga maksimum atau minimum, tergantung pada tanda sinyal kesalahan penggerak apakah positif atau negatif. Persamaan untuk kontrol ini dapat dituliskan sebagai berikut :

$$m(t) = M_1 \dots\dots\dots e(t) > 0 \quad \dots\dots(2-12)$$

$$m(t) = M_2 \dots\dots\dots e(t) < 0 \quad \dots\dots(2-13)$$

dimana M_1 dan M_2 merupakan konstanta. Diagram blok untuk kontroler dua posisi ditunjukkan pada gambar 2.7.



GAMBAR 2.7

DIAGRAM BLOK KONTROLER DUA POSISI

II.5.2. KONTROL PROPORSIONAL PLUS INTEGRAL PLUS DERIVATIF

Gabungan aksi kontrol proporsional, kontrol integral dan kontrol derivatif membentuk aksi kontrol proporsional plus integral plus derivatif, atau yang dikenal dengan kontrol PID.

Pada desain sistem kontrol analog banyak digunakan kontroler PID karena tipe ini mempunyai kestabilan dan performance yang baik.

Persamaan kontroler PID ini adalah sebagai berikut :

$$P = K_p \cdot E_p + K_p \cdot K_i \int_0^t E_p dt + K_p \cdot K_d \cdot \frac{dE_p}{dt} + P_i(0) \quad \dots (2-14)$$

Dan persamaan fungsi alihnya adalah :

$$\frac{U(S)}{E(S)} = D(S) = K_p + \frac{K_i}{S} + K_d \cdot S \quad \dots (2-15)$$

dimana :

P = output kontroler
 E_p = error proses
 K_p = konstanta proporsional
 K_i = konstanta integral
 K_d = konstanta derivatif
 $P_i(0)$ = harga awal output kontroler

Persamaan 2-14 adalah persamaan kontroler PID untuk waktu kontinyu. Pada penerapan kontroler PID secara digital maka yang berperan adalah waktu diskrit. Oleh karena itu persamaan 2-14 di atas harus diubah menjadi persamaan differensial diskrit. Untuk itu, mula-mula dilakukan penurunan terhadap kedua sisi persamaan tersebut sebagai berikut :

$$\begin{aligned}\frac{dP}{dt} &= K_p \cdot \frac{dE_p}{dt} + K_p \cdot K_i \cdot \frac{d}{dt} \left(\int E_p dt \right) + K_p \cdot K_d \cdot \frac{d^2 E_p}{dt^2} \\ &= K_p \cdot \frac{dE_p}{dt} + K_p \cdot K_i \cdot E_p + K_p \cdot K_d \cdot \frac{d}{dt} \left[\frac{dE_p}{dt} \right]\end{aligned}$$

Persamaan di atas memberikan berapa banyak perubahan output untuk tiap perubahan input dalam selang waktu yang mendekati nol, yaitu dt . Tetapi didalam sistem komputer, sampling dilakukan untuk tiap interval waktu tertentu, yaitu T . Oleh karena itu, persamaan harus diubah dengan menggunakan selang waktu T , dan dE_p diubah menjadi ΔE_p , dimana ΔE_p adalah perubahan E_p selama selang waktu T . Persamaan sekarang menjadi :

$$\frac{\Delta P}{T} = K_p \cdot \frac{\Delta E_p}{T} + K_p \cdot K_i \cdot E_p + K_p \cdot K_d \cdot \frac{\Delta}{T} \left[\frac{\Delta E_p}{T} \right]$$

atau :

$$\Delta P = K_p \cdot \Delta E_p + K_p \cdot K_i \cdot E_p + K_p \cdot K_d \cdot \Delta \left[\frac{\Delta E_p}{T} \right]$$

Untuk mengganti ΔP dengan P , maka persamaan tersebut harus diubah dengan melihat P sebagai selisih antara nilai yang sekarang dengan nilai yang diambil sebelumnya. Hal yang sama juga berlaku untuk error.

$$\Delta P = P[n] - P[n-1]$$

$$\Delta E_p = E_p[n] - E_p[n-1]$$

Dengan demikian persamaan menjadi :

$$P[n] - P[n-1] = K_p (E_p[n] - E_p[n-1]) + K_p \cdot K_i \cdot E_p \cdot T + \frac{K_p \cdot K_d}{T} (\Delta E_p[n] - \Delta E_p[n-1]) \quad \dots (2-16)$$

Pada suku terakhir persamaan 2-16, $\Delta E_p[n]$ dan $\Delta E_p[n-1]$

dapat dikembangkan lagi sebagai berikut :

$$\Delta E_p[n] = E_p[n] - E_p[n-1]$$

$$\Delta E_p[n-1] = E_p[n-1] - E_p[n-2]$$

Dengan melakukan substitusi ke persamaan 2-16, didapat persamaan berikut :

$$P[n] = P[n-1] + K_p(E_p[n] - E_p[n-1]) + K_p.K_i.E_p[n].T + \frac{K_p.K_d}{T}(E_p[n] - 2.E_p[n-1] + E_p[n-2]) \quad \dots (2-17)$$

Dari persamaan di atas, maka kontroler PID dapat diimplementasikan dengan software untuk aplikasi ke komputer.

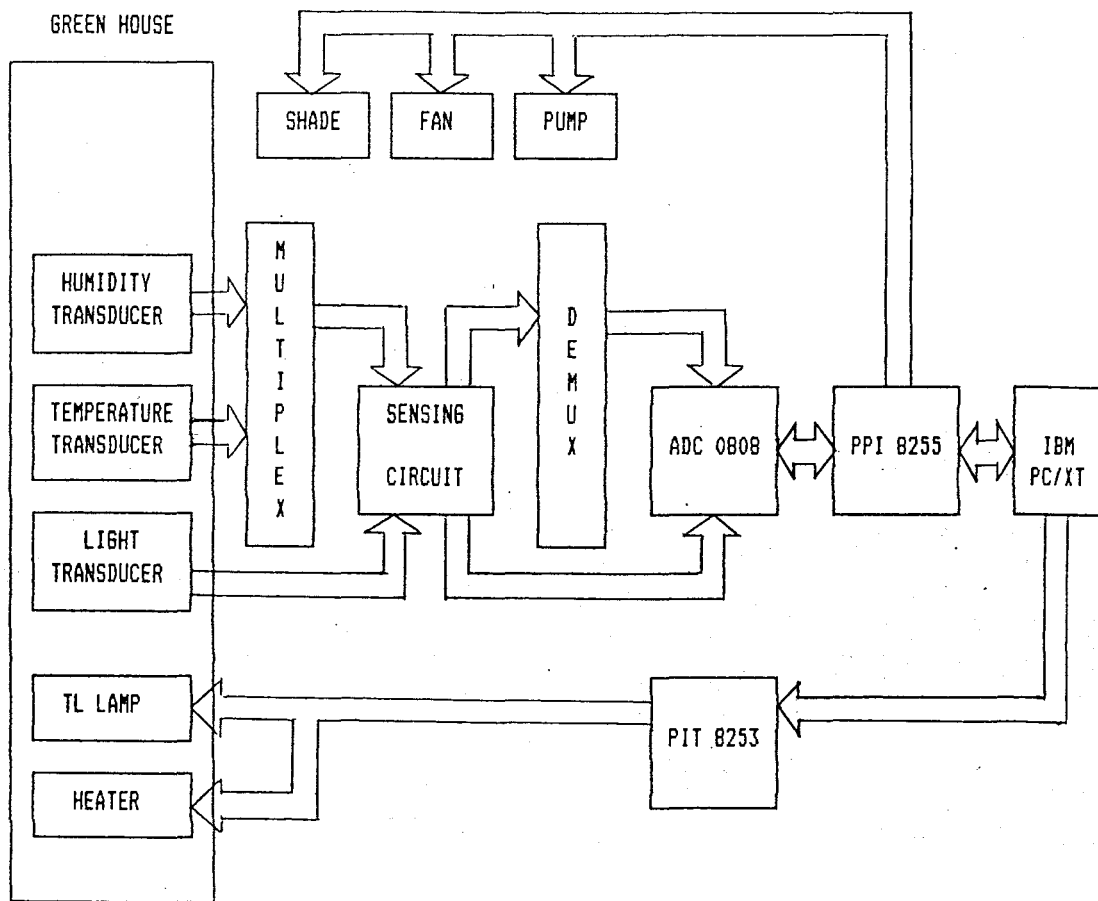
BAB III

PERENCANAAN DAN PEMBUATAN

III.1. UMUM

Uraian perencanaan alat ini meliputi penjelasan unjuk kerja sistem secara keseluruhan berdasar blok diagram alat yang dibuat, kemudian dilanjutkan dengan uraian mengenai perencanaan perangkat keras mekanik berupa perencanaan rumah kaca dan perhitungan termodinamika untuk air conditioning, perencanaan perangkat keras elektronik serta perencanaan perangkat lunak.

Blok diagram dari alat yang dibuat ditunjukkan pada gambar 3.1. Akuisisi data berlangsung dari modul pengindra sensor yang mengubah besaran fisis, yaitu temperatur, kelembaban relatif dan intensitas cahaya, ke level tegangan analog. Keluaran tegangan analog dari transduser ini kemudian mengalami pemrosesan sinyal analog berupa pengkondisian sinyal, penguatan dan pemfilteran. Pemrosesan ini berlangsung dalam blok sensing circuit. Selanjutnya sinyal analog hasil pemrosesan diubah dalam bentuk digital oleh modul Analog to Digital Converter. Pemrosesan secara digital oleh komputer dilakukan dengan perantaraan I/O interface berupa PPI 8255 dan PIT 8253 yang menghubungkan rangkaian luar dengan komputer.



GAMBAR 3.1

BLOK DIAGRAM PERANGKAT KERAS SISTEM

Acuan sistem berupa nilai besaran yang ingin dicapai dan setting peralatan pendukung pada software yang dibuat dimasukkan dengan perantara keyboard. Kemudian alur kontrol pada program akan mengecek pengadaaan aksi kontrol ke sistem. Peralatan pendukung seperti pompa (pump) dan fan akan diaktifkan apabila diset 'on' yaitu bila difungsikan untuk pemupukan, penyiraman dan blower. Sedangkan penyesuaian data hasil sample dengan setting

besaran yang dikontrol dilakukan dengan aksi kontrol berupa umpan balik ke sistem oleh rangkaian driver. Heater diaktifkan untuk penyesuaian besaran temperatur, dimmer lampu flourescent dan gerak motor stepper untuk peneduh (shade) diperlukan dalam penyesuaian besaran intensitas cahaya, sedangkan pengaktifan pompa dan fan untuk efek penyemprotan air menjadi butiran halus (sprayer) adalah untuk penyesuaian besaran kelembaban relatif. Adapun penggunaan rangkaian multiplexing dan demultiplexing dimaksudkan untuk meminimisasi rangkaian karena pengkondisian sinyal transduser yang sama.

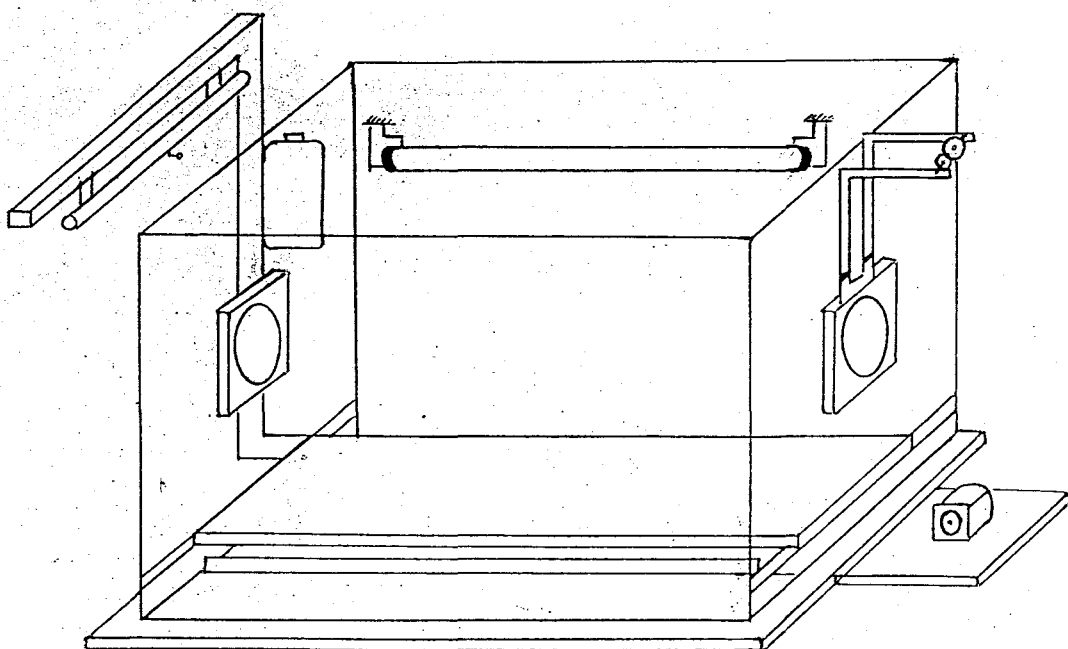
III.2. PERENCANAAN RUMAH KACA (GREEN HOUSE)

III.2.1. KONSTRUKSI RUMAH KACA

Model rumah kaca yang direncanakan, seperti yang ditunjukkan pada gambar 3.2, mempunyai dimensi 80 cm x 50 cm x 50 cm, dari bahan kaca dengan ketebalan 5 mm.

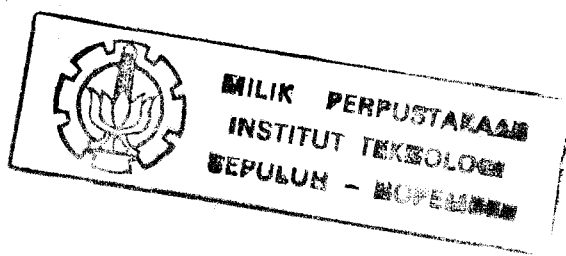
Perencanaan konstruksi model rumah kaca tersebut didasarkan atas beberapa pertimbangan. Penggunaan psychrometer untuk konversi besaran kelembaban relatif memerlukan adanya aliran udara yang melintasi wet bulb. Untuk mendukung keperluan ini maka pada sisi kiri dan kanan model rumah kaca ditempatkan fan. Pada sisi kiri, arah aliran udaranya masuk sedang pada sisi kanan aliran udaranya keluar. Konstruksi ini juga berfungsi sebagai

penyuplai angin untuk tanaman anggrek yang dibudidayakan dan sebagai peralatan pendukung pengaturan kelembaban ruangan yaitu dengan mengubah semprotan air dari nozzle yang diletakkan di samping fan menjadi butiran-butiran halus.



GAMBAR 3.2

KONSTRUKSI MODEL RUMAH KACA



Pemanas yang diperlukan untuk menaikkan suhu dalam model rumah kaca ditempatkan pada bagian alas. Di atasnya ditempatkan lembaran kaca. Fungsinya yaitu untuk mencegah adanya aliran air yang mengenai pemanas. Sedangkan penyebaran panas sendiri tidak terganggu karena adanya celah yang memungkinkan daya apung panas secara alami.

Aliran air sisa penyiraman ataupun pengaturan kelembaban ditampung dalam lekukan pada dasar model rumah kaca yang difungsikan sebagai saluran air. Pada ujungnya dibuatkan lubang untuk pembuangan keluar.

Pada bagian atas dari model rumah kaca dibuatkan tempat untuk menggantung tanaman anggrek. Sedang pada bagian bawah atapnya, ditempatkan lampu fluorescent untuk mendukung pencahayaan pada sistem.

Kemudian untuk keringkasan model yang dibuat, pada bagian luar dibuatkan juga sandaran untuk menempatkan penampung air, pompa, motor stepper dan lapisan peneduh. Penampung air dan pompa digunakan untuk mendukung fasilitas penyiraman dan pemupukan dalam sistem.

III.2.2. PERENCANAAN PEMANAS

Elemen pemanas diperlukan untuk menghasilkan panas sehingga dapat menaikkan suhu ruangan. Dalam hal penentuan besar pemanas, perlu diperhatikan adanya panas yang hilang diakibatkan proses perpindahan panas yang

terjadi akibat konduksi dan konveksi. Idealnya, pemanas yang direncanakan harus lebih besar daripada jumlah antara panas yang digunakan untuk menaikkan volume udara dalam ruangan sebesar temperatur maksimum yang direncanakan dengan kehilangan panas yang disebabkan oleh konveksi dan konduksi. Hal ini akan menjamin pencapaian temperatur maksimum yang direncanakan.

Dimisalkan temperatur kamar adalah $T_1 = 27^{\circ}\text{C} = 80.6^{\circ}\text{F}$. Sedangkan temperatur maksimum dalam rumah kaca yang direncanakan adalah sebesar $T_2 = 30^{\circ}\text{C} = 86^{\circ}\text{F}$, yaitu sesuai dengan temperatur maksimum budidaya anggrek. Seiring pemanasan pada ruangan maka terjadi proses konduksi pada kaca dan konveksi dari kaca ke udara luar. Karena beda temperatur yang cukup kecil antara ruang dalam rumah kaca dengan luar, diambil asumsi bahwa temperatur pada kaca adalah uniform pada $T_3 = 28.5^{\circ}\text{C} = 83.3^{\circ}\text{F}$. Jadi beda temperatur $\Delta t = 83.3^{\circ}\text{F} - 80.6^{\circ}\text{F} = 2.7^{\circ}\text{F}$.

Dengan demikian panas yang hilang akibat proses konveksi dapat dihitung sebagai berikut :

- a. Panas yang dipindahkan pada dinding vertikal bagian depan dan belakang

Dimensi permukaan = $80 \text{ cm} \times 50 \text{ cm} = 2.62 \text{ ft} \times 1.64 \text{ ft}$

Harga L didapat dari persamaan 2-11, yaitu :

$$L = \frac{L_{\text{horiz}} \cdot L_{\text{vert}}}{L_{\text{horiz}} + L_{\text{vert}}} = \frac{2.62 \times 1.64}{2.62 + 1.64} = 1 \text{ ft}$$

Dan :

$$L^3 \cdot \Delta t = 2.7$$

Dengan harga di atas, koefisien konveksinya dicari dari persamaan 2-4. Jadi :

$$h_m = 0.28 \times (2.7)^{1/4}$$

$$\Rightarrow h_m \approx 0.36 \text{ Btu/h.ft}^2 \cdot ^\circ\text{F}$$

Maka besarnya panas yang dipindahkan menurut persamaan 2-3 adalah :

$$q_1 = 0.36 \times (2.62 \times 1.64) \times 2.7$$

$$\therefore q_1 \approx 4.18 \text{ Btu/h}$$

b. Panas yang dipindahkan pada dinding vertikal bagian kiri dan kanan

$$\text{Dimensi permukaan} = 50 \text{ cm} \times 50 \text{ cm} = 1.64 \text{ ft} \times 1.64 \text{ ft}$$

Maka :

$$L = 0.82 \text{ ft}$$

$$L^3 \cdot \Delta t = 1.49$$

Dengan harga di atas, koefisien konveksinya dicari dari persamaan 2-4. Jadi :

$$h_m = 0.28 \times (2.7 / 0.82)^{1/4}$$

$$\Rightarrow h_m \approx 0.38 \text{ Btu/h.ft}^2 \cdot ^\circ\text{F}$$

Maka besarnya panas yang dipindahkan adalah :

$$q_2 = 0.38 \times (1.64 \times 1.64) \times 2.7$$

$$\therefore q_2 \approx 2.76 \text{ Btu/h}$$

c. Panas yang dipindahkan pada dinding atas

Dimensi permukaan = 80 cm x 50cm = 2.62 ft x 1.64 ft

Maka :

$$L = 1 \text{ ft}$$

$$L^3 \Delta t = 2.7$$

Dengan harga di atas, koefisien konveksinya dicari dari persamaan 2-8. Jadi :

$$h_m = 0.27 \times (2.7)^{1/4}$$

$$\Rightarrow h_m \approx 0.35 \text{ Btu/h.ft}^2 \cdot ^\circ\text{F}$$

Maka besarnya panas yang dipindahkan adalah :

$$q_3 = 0.35 \times (2.62 \times 1.64) \times 2.7$$

$$\therefore q_3 \approx 4.06 \text{ Btu/h}$$

Jadi panas total yang hilang pada seluruh permukaan sebesar :

$$q_k = q_1 + q_2 + q_3 = 4.18 + 2.76 + 4.06$$

$$\therefore q_k = 11 \text{ Btu/h}$$

Selanjutnya akan diperhitungkan besarnya panas yang dibutuhkan untuk menaikkan temperatur udara dalam rumah kaca. Dalam hal ini paling tidak harus diketahui besarnya kelembaban relatif udara luar dan kelembaban relatif (atau temperatur wet bulb) dalam rumah kaca setelah pemanasan. Dimisalkan pengukuran dengan termohigrometer di udara luar pada 86°F DB adalah 80%. Dari psychrometric chart dapat diketahui besarnya temperatur wet bulb yaitu 74°F WB dan entalpi atau kandungan panas totalnya sebesar

38.1 Btu/lb. Kemudian adanya pemanasan dalam rumah kaca untuk pencapaian temperatur sampai 86°F DB, menyebabkan penunjukan kelembaban relatif 70% atau pengukuran wet bulbnya adalah 75.6°F. Jadi entalpi atau kandungan panas total setelah pemanasan menjadi 39.7 Btu/lb (dibaca dari psychrometric chart). Dengan demikian panas yang diperlukan untuk menaikkan temperatur dari 80.6°F menjadi 86°F = $39.7 - 38.1 = 1.6$ Btu/lb.

Untuk mendapatkan besar panas yang perlu ditambahkan per satuan waktu maka harus diketahui berat udara yang masuk ke sistem. Berat udara yang masuk ini dapat dicari dari volume udara yang masuk dikalikan dengan kerapatan udaranya. Detail perhitungannya adalah sebagai berikut :

- Pertama harus diketahui kecepatan udara yang masuk yang disuplai dengan fan. Pengukuran dengan airflow meter didapat kecepatan aliran udara = 3.86 m/sec. Volume udara yang masuk dihitung dari hasil kali kecepatan aliran tersebut dengan luas penampang fan (diameter fan = 12 cm). Didapat :

$$\begin{aligned} V_{udara} &= 3.86 \times \pi \times (0.06)^2 \\ &= 0.044 \text{ m}^3/\text{sec} \end{aligned}$$

Karena $1 \text{ m}^3 = 35.31 \text{ ft}^3$, maka :

$$\therefore V_{udara} = 5593.1 \text{ ft}^3/\text{h}$$

- Kondisi udara luar adalah 86°F DB dan 74°F WB. Dari psychrometric chart didapat spesifik volumenya sebesar

13.79 ft³/lb. Jadi kerapatannya = 1/13.79 lb/ft³.

Maka :

$$\text{berat udara} = 5593.1 / 13.79 = 405.6 \text{ lb.}$$

Pada pemanasan, volume udara akan bertambah tetapi beratnya tetap. Oleh karena panas yang diperlukan untuk menaikkan temperatur udara dalam rumah kaca dari 27°C ke 30°C adalah :

$$q_{\text{heat}} = 405.6 \times 1.6 \approx 649 \text{ Btu/h}$$

Jadi total panas yang diperlukan :

$$q = q_k + q_{\text{heat}} = 11 + 649 = 660 \text{ Btu/h}$$

Dalam satuan SI (1 Btu/h = 0.29 J/s) :

$$\therefore q \approx 193 \text{ J/s}$$

Dari perhitungan sebelumnya, secara teoritis besarnya pemanas harus lebih besar dari panas yang diperlukan untuk menjamin pencapaian temperatur maksimum. Pada sistem yang direncanakan, daya pemanas yang dipakai adalah sebesar 350 Watt. Hal ini dimaksudkan agar pemanasan dapat berlangsung lebih cepat dan dapat mencapai temperatur maksimum yang lebih tinggi. Dari perhitungan di atas, lamanya waktu pemanasan bila digunakan pemanas dengan daya 350 Watt adalah $193/350 = 0.55$ detik. Tetapi dalam kenyataannya, waktu yang diperlukan tidaklah secepat itu sebab diperlukan juga waktu untuk penyebaran panas, perpindahan panas dan pemanasan sumber pemanas itu sendiri.

III.3. PERENCANAAN HARDWARE

Pada perencanaan perangkat keras elektronik ini, sistem dibagi atas beberapa modul yaitu :

1. Modul Interface, meliputi :

- Rangkaian card Interface yang terdiri dari I/O interface yaitu rangkaian PPI 8255 dan PIT 8253, rangkaian decoder serta buffer.
- Rangkaian I/O buffer.

2. Modul Pengindra sensor, meliputi :

- Rangkaian multiplexer dan demultiplexer.
- Rangkaian detektor temperatur.
- Rangkaian detektor kelembaban relatif.
- Rangkaian detektor intensitas cahaya.

3. Modul Analog to Digital Converter berupa rangkaian ADC 0808.

4. Modul Driver untuk umpan balik sistem, meliputi :

- Rangkaian pengaktif pompa pengisap.
- Rangkaian penggerak motor stepper.
- Rangkaian penggerak fan.
- Rangkaian pengatur pemanas.
- Rangkaian dimmer lampu fluorescent (TL).

III.3.1. MODUL INTERFACE

Modul interface ini berfungsi untuk menghubungkan komputer dengan peralatan luar. Pada perencanaan, modul

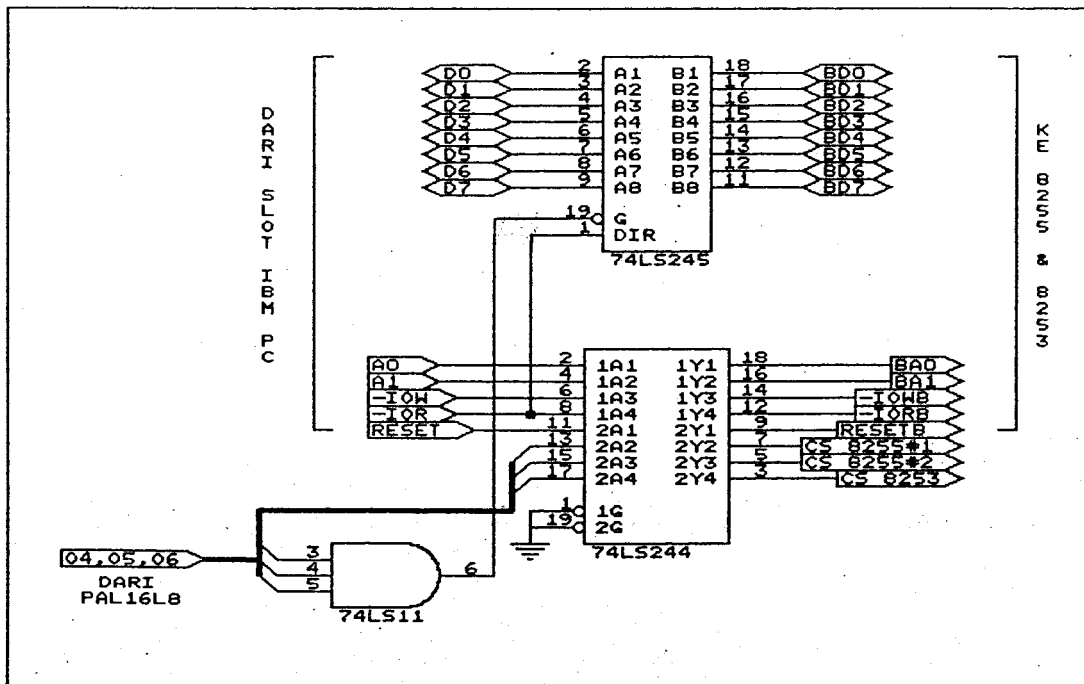
ini dibagi atas dua bagian yaitu interface card dan I/O buffer. Interface card terdiri atas rangkaian buffer untuk data, decoder, PPI 8255 dan PIT 8253, sedangkan I/O buffer berisikan IC buffer schmitt trigger 74LS244.

III.3.1.1. RANGKAIAN BUFFER

Pada interface card, bus data dari komputer dibuffer dengan bidirectional buffer 74LS245. Hal ini dimaksudkan untuk memperkuat sinyal data dari komputer dan juga untuk mengisolasi sistem data bus terhadap peralatan luar pada saat peralatan luar tersebut tidak diaktifkan.

Sistem data bus komputer dihubungkan pada sisi A, sedang data bus peralatan pada sisi B IC 74LS245. Pin enable (\overline{G}) dihubungkan dengan keluaran IC 74LS11 yang pada saat logik 0 akan menghubungkan data bus dengan peripheral, pada perencanaan yaitu PPI 8255 dan PIT 8253. Keluaran IC 74LS11 (Dual AND gate) akan berlogik 0 bila salah satu inputnya berlogik 0. Keadaan ini dimungkinkan dengan menghubungkan ketiga inputnya dengan keluaran decoder yang berlogik 0 saat mengaktifkan peripheral.

Arah aliran data ditentukan oleh pin DIR, yang mana bila berlogik 0 aliran datanya dari sisi B ke sisi A (saat I/O port read bus cycle) dan sebaliknya bila berlogik 1 (saat I/O port write cycle). Untuk keperluan ini pin DIR dihubungkan ke sinyal kontrol $\overline{I/O}$ komputer.

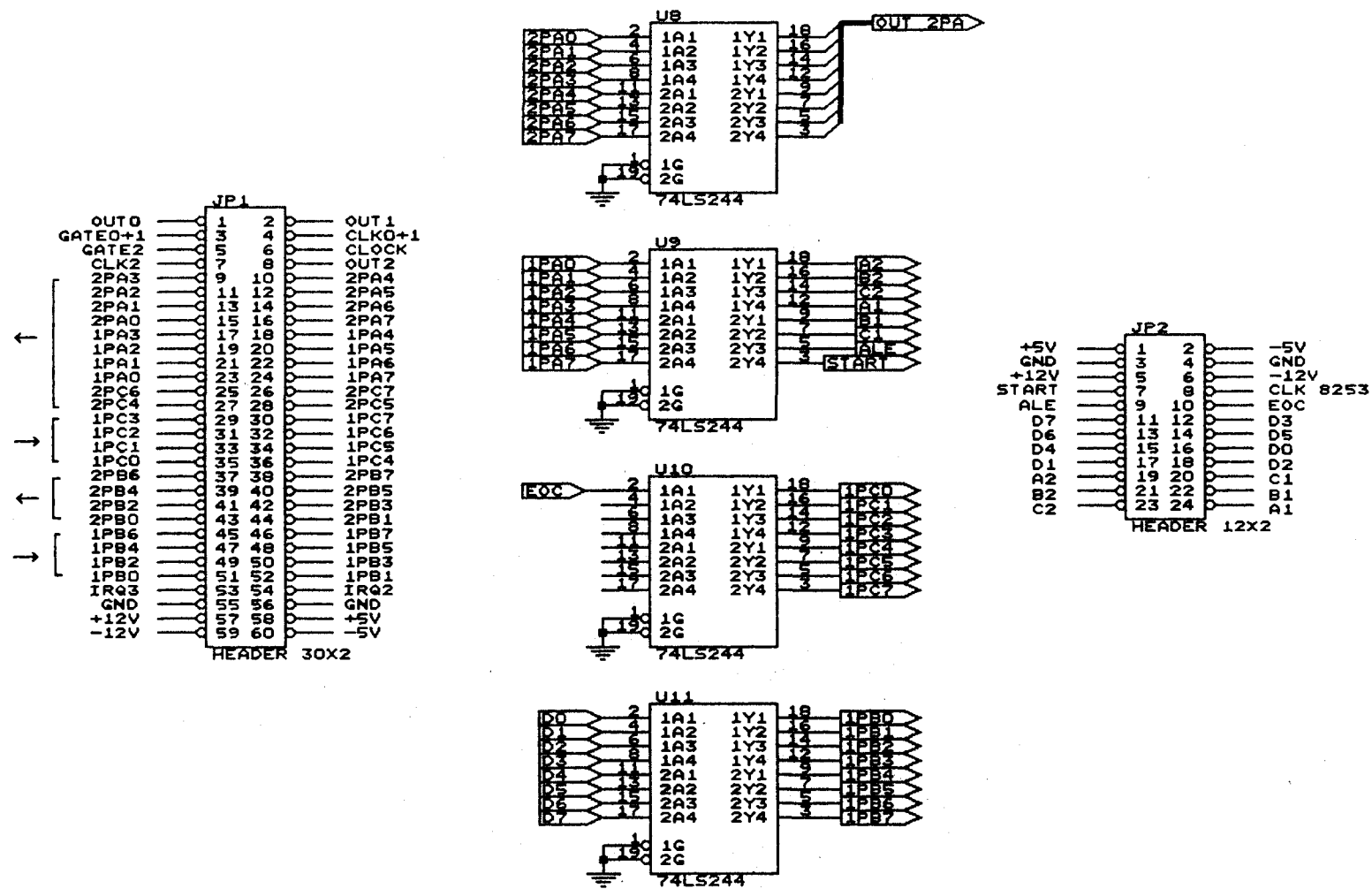


GAMBAR 3.3

RANGKAIAN BUFFER PADA INTERFACE CARD

Beberapa sinyal kontrol yang dihubungkan ke peripheral juga dibuffer untuk memperkuat sinyal. Untuk ini cukup digunakan unidirectional buffer 74LS244.

Selain rangkaian buffer pada interface card, direncanakan juga rangkaian I/O buffer yang menggunakan IC 74LS244. Buffer ini menghubungkan peripheral (PPI 8255 dan PIT 8253) dengan peralatan luar. Selain berfungsi memperkuat arus keluaran peripheral, rangkaian buffer juga berperan sebagai unit penghubung dan distributor sinyal. Rangkaian I/O buffer ditunjukkan pada gambar 3.4.



GAMBAR 3.4
MODUL BUFFER I/O

III.3.1.2. RANGKAIAN DECODER

Rangkaian decoder ini digunakan untuk pengaktifan I/O device yang dipakai yaitu PPI 8255 dan PIT 8253. Dalam perencanaan ini digunakan IC PAL 16L8 yang merupakan programmable input/output guna keperluan decoding.

PAL 16L8 memiliki 10 input yang terletak pada pin nomer 1-9 dan satu lagi terletak pada pin nomer 11. Pin khusus untuk output adalah pin 12 dan pin 19, sedangkan pin 13-18 dapat berfungsi sebagai input ataupun output. Output dari IC PAL16L8 ini adalah aktif 'low'.

Diagram logika dari PAL16L8 berupa pola-pola sekering yang diatur sedemikian rupa sehingga berbentuk matrik. Baris vertikal diberi nomer 0-31 dan merupakan jalur input. Baris horisontal diberi nomer 0-64 yang merupakan product term dalam pemrogramannya. Setiap product term dikelompokkan dan berisi 8 product line, dimana 7 product line dibagian bawah dijumlahkan (OR) dan satu dibagian atas mengendalikan buffer inverter pada output OR gate. Dengan demikian jumlah sekering yang ada adalah 2048 buah dan dengan menentukan sekering mana yang diputus dan mana yang tidak maka akan didapat suatu fungsi logika.

Untuk implementasi rangkaian maka harus ditentukan terlebih dahulu persamaan logika untuk decoding. Adapun persamaan logika yang diperlukan tersebut akan diturunkan

dari tabel 3-1 yang menunjukkan alamat dari I/O device yang digunakan dalam perencanaan.

Dari tabel kebenaran, bit A0 dan A1 dihubungkan ke pin A0 dan A1 dari masing-masing I/O device dan digunakan untuk memilih port A (PA), port B (PB), port C (PC) dan Control word register (CW) dari PPI 8255 ataupun counter 0 (CTR0), counter 1 (CTR1), counter 2 (CTR2) dan control word register (CW) dari PIT 8253. Sedangkan bit-bit A2-A9 digunakan untuk membentuk fungsi logika yang mengaktifkan chip select dari masing-masing device.

Karena chip select aktif 'low' maka fungsi logika yang didapat dari tabel kebenaran dalam bentuk sum of product adalah sebagai berikut :

$$\overline{CS} \text{ 8255\#1} = \overline{A9.A8.A7.A6.A5.A4.A3.A2.AEN} \quad \dots (3-1)$$

$$\overline{CS} \text{ 8255\#2} = \overline{A9.A8.A7.A6.A5.A4.A3.A2.AEN} \quad \dots (3-2)$$

$$\overline{CS} \text{ 8253} = \overline{A9.A8.A7.A6.A5.A4.A3.A2.AEN} \quad \dots (3-3)$$

Penambahan masukan berupa sinyal AEN (Address enable) digunakan untuk mengaktifkan tri-state buffer pada output. Sinyal AEN adalah aktif 'high' dan akan high apabila sistem board melakukan proses DMA ke I/O device, sebaliknya akan 'low'. Pada saat 'low' inilah keluaran inverting dari pin input AEN pada IC PAL akan 'high' dan meng-enable tri-state buffer. Untuk jelasnya, diagram logika dan implementasi fungsi decoding diatas digambar-

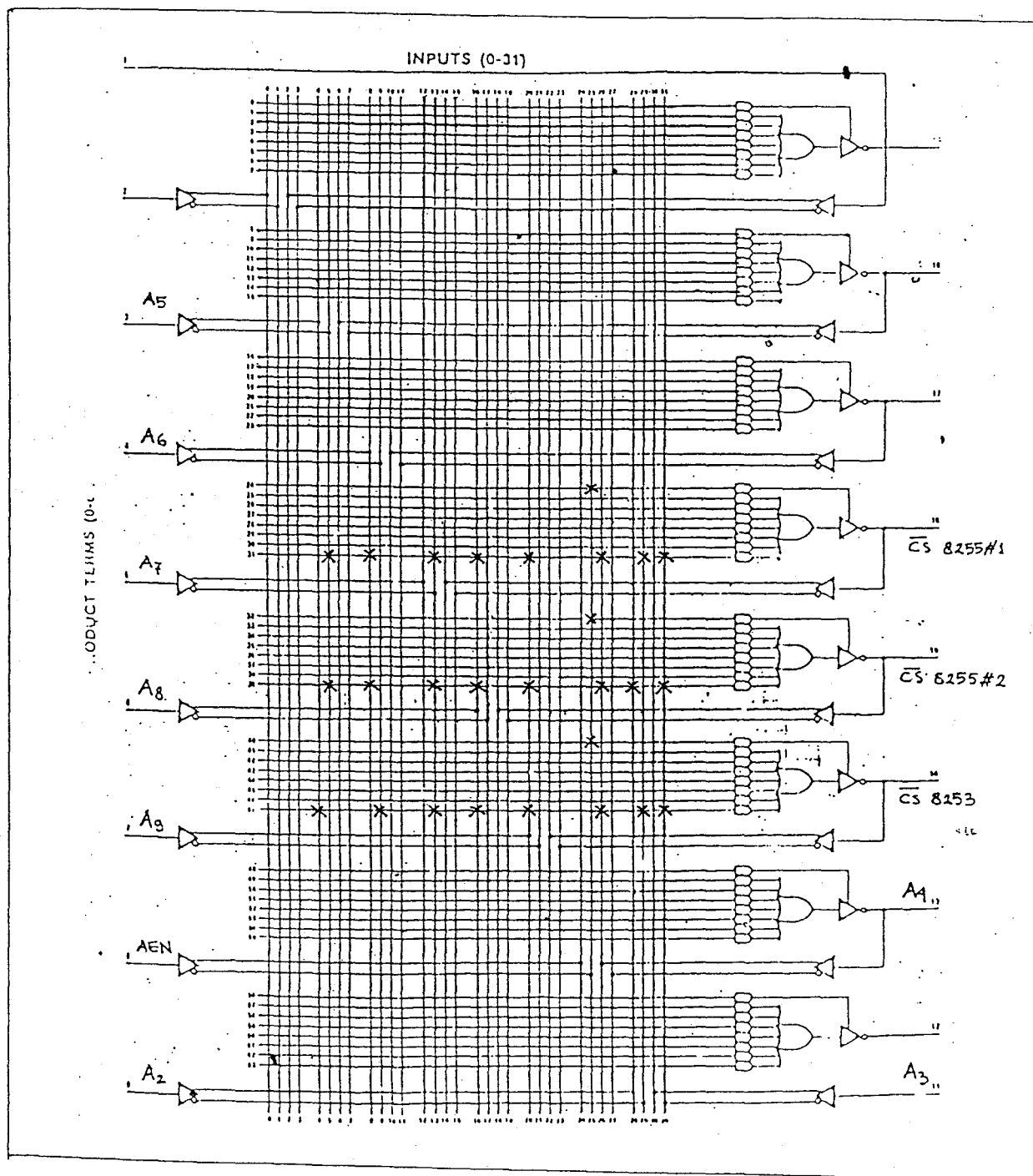
kan pada gambar 3.5 dan rangkaian dekodernyua ditunjukkan pada gambar 3.6.

Pada gambar 3.5, tanda 'x' menyatakan sekring dalam keadaan terhubung. Terlihat bahwa pin 3-9, pin 11 dan 13 digunakan sebagai input sedangkan pin output adalah pin 14 untuk mengaktifkan 8253, pin 15 untuk 8255#2 dan pin 16 untuk 8255#1.

TABEL 3-1.

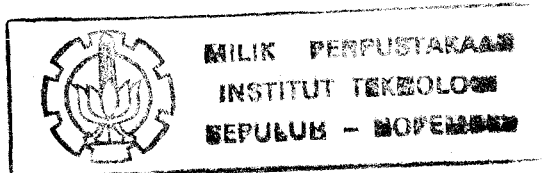
TABEL KEBENARAN PENGALAMATAN I/O PORT

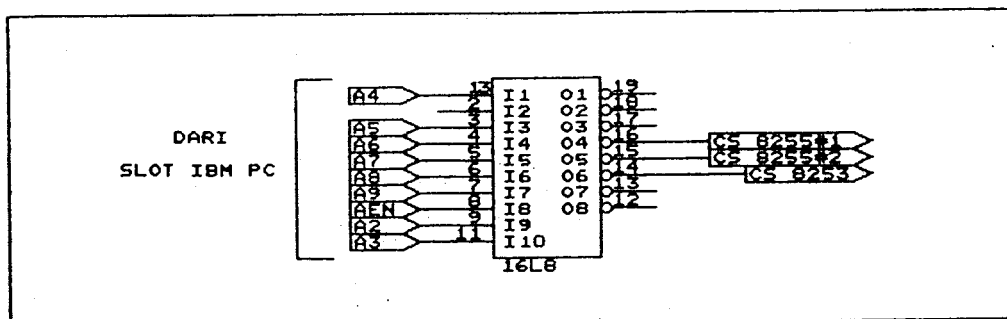
Port	Alamat	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
PA 8255#1	350H	1	1	0	1	0	1	0	0	0	0
PB 8255#1	351H	1	1	0	1	0	1	0	0	0	1
PC 8255#1	352H	1	1	0	1	0	1	0	0	1	0
CW 8255#1	353H	1	1	0	1	0	1	0	0	1	1
PA 8255#2	354H	1	1	0	1	0	1	0	1	0	0
PB 8255#2	355H	1	1	0	1	0	1	0	1	0	1
PC 8255#2	356H	1	1	0	1	0	1	0	1	1	0
CW 8255#2	357H	1	1	0	1	0	1	0	1	1	1
CTR0 8253	330H	1	1	0	0	1	1	0	0	0	0
CTR1 8253	331H	1	1	0	0	1	1	0	0	0	1
CTR2 8253	332H	1	1	0	0	1	1	0	0	1	0
CW 8253	333H	1	1	0	0	1	1	0	0	1	1



GAMBAR 3.5

Diagram Logika fungsi I/O decoding dengan PAL16L8





GAMBAR 3.6

RANGKAIAN DEKODER DENGAN PAL16L8

III.3.1.3. RANGKAIAN PPI 8255

Untuk mendukung kontrol data dan pengaturan sistem, seperti lintas data dari dan ke ADC serta pengaktifan rangkaian-rangkaian driver, diperlukan dua buah PPI 8255. Hubungan rangkaiannya adalah seperti yang diperlihatkan pada gambar 3.7. Kedua PPI 8255 tersebut dioperasikan pada mode 0 yaitu mode input - output.

Adapun penggunaan dari masing-masing PPI tersebut adalah sebagai berikut :

a. PPI 8255#1

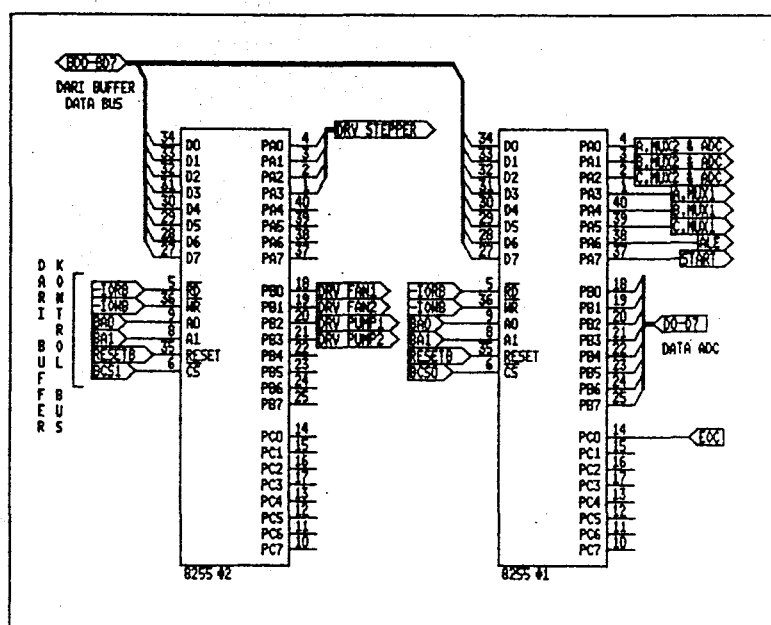
- Port A (1PA₀ - 1PA₇) digunakan sebagai port output yaitu dihubungkan ke pin-pin kontrol ADC 0808, multiplexer dan demultiplexer.
- Port B (1PB₀ - 1PB₇) digunakan sebagai port input yaitu untuk mengambil data hasil konversi ADC 0808.
- Port C_{lower} (1PC₀) digunakan sebagai port input

yaitu untuk mendeteksi sinyal EOC (End of Conversion) dari ADC 0808.

- Port Copper diprogram sebagai port output tetapi tidak dihubungkan ke peralatan.

b. PPI 8255#2

- Port A (2PA₀ - 2PA₄) digunakan sebagai port output yaitu untuk driver motor stepper.
- Port B (2PB₀ - 2PB₄) digunakan sebagai port output yaitu untuk driver fan dan pompa pengisap.
- Port C diprogram sebagai port output tetapi tidak dihubungkan ke peralatan.



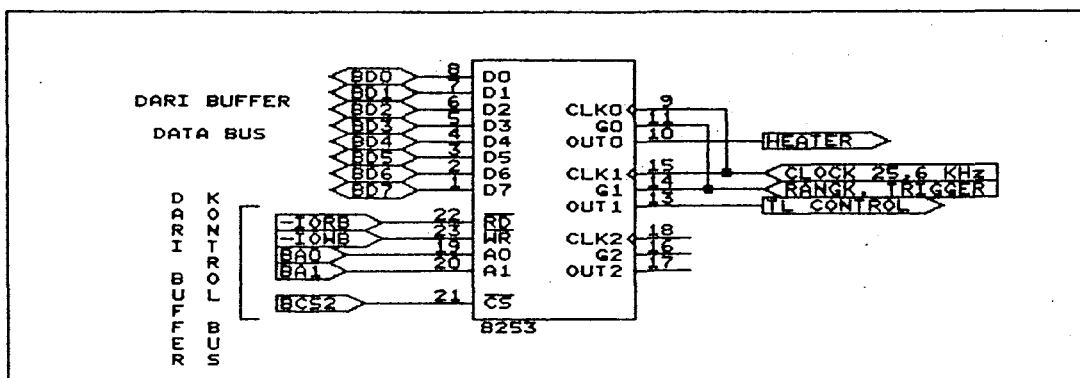
GAMBAR 3.7

RANGKAIAN PPI 8255

III.3.1.4. RANGKAIAN PIT 8253

Rangkaian PIT 8253 ini digunakan sebagai pengontrol sistem secara kontinyu. Pada perencanaan sistem, pengaturan temperatur dengan pemanas dan pengaturan intensitas cahaya dengan rangkaian dimmer lampu fluorescent menggunakan kontrol kontinyu. Dengan demikian rangkaian-rangkaian itulah yang dikontrol oleh keluaran PIT 8253.

Untuk mendukung fungsi di atas, maka PIT 8253 dioperasikan pada mode 1 yaitu *Programmable One Shot*. Pin-pin CLK, GATE dan OUT dari Counter 0 dan Counter 1 (Counter 2 tidak digunakan) dihubungkan seperti gambar 3.8. Input berupa sinyal clock dan trigger disesuaikan dengan rangkaian driver dalam kaitannya dengan pemakaian tegangan AC untuk men-drive pemanas dan lampu.



GAMBAR 3.8

RANGKAIAN PIT 8253

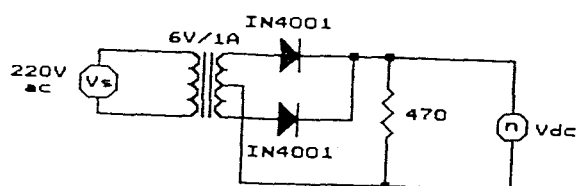
RANGKAIAN TRIGGER

Rangkaian trigger ini menghasilkan pulsa trigger yang menentukan awal durasi one-shot. Pada perencanaannya, pertimbangan didasarkan atas diperlukannya pulsa trigger dalam level TTL yang sinkron dengan tegangan jala-jala. Rangkaian seperti yang ditunjukkan pada gambar 3.9 dibuat untuk tujuan tersebut.

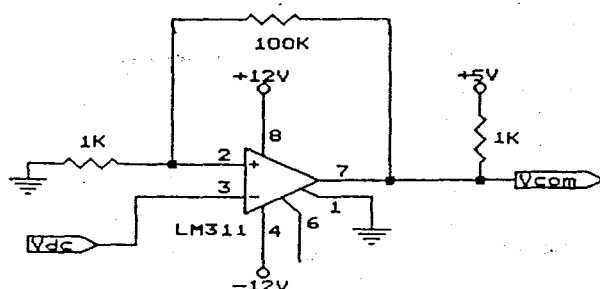
Gambar 3.9(a) merupakan penyearah gelombang penuh yang mengubah tegangan jala-jala 220 V menjadi tegangan searah dengan tegangan rata-rata sekitar 4,8 V. Tegangan ini kemudian diinputkan ke rangkaian detektor penyilang nol (zero crossing detector), seperti tampak pada gambar 3.9(b). Rangkaian detektor penyilang nol ini dirancang dengan menggunakan IC LM311 yang memang dioptimalkan pada pemakaian detektor taraf tegangan. Tahanan 100K dan 100 Ohm memberikan histerisis sekitar 20 mV untuk meredam noise. Tahanan pull-up 1K juga dihubungkan pada keluaran untuk menahan keluaran dalam level TTL. Sinyal trigger yang dihasilkan terlihat pada gambar 3.9(c).

CLOCK PIT 8253

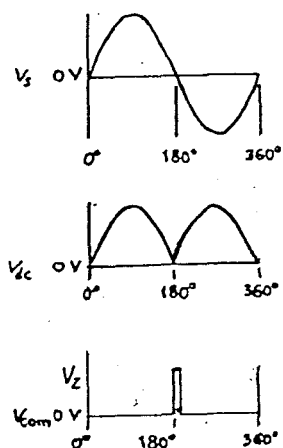
Sinyal clock ini menentukan durasi keluaran PIT 8253 yaitu bergantung banyaknya sampling yang dilakukan clock



(a) Rangkaian penyearah gelombang penuh



(b) Rangkaian detektor penyilang nol



(c) Sinyal trigger

GAMBAR 3.9

RANGKAIAN DAN SINYAL TRIGGER

terhadap gelombang sinus jala-jala listrik. Direncanakan bahwa satu periode gelombang hasil penyearahan disampling sebanyak 256 sample, maka perhitungan frekuensi clock yang dibuat adalah sebagai berikut :

- periode sinyal hasil penyearahan rangkap :

$$T_s = \frac{1}{2 \times f_{\text{jala-jala}}} = \frac{1}{100} = 10 \text{ ms}$$

maka :

$$f_{\text{clock}} = \frac{1}{\frac{T_s}{256}} = \frac{256}{T_s} = 25,6 \text{ KHz}$$

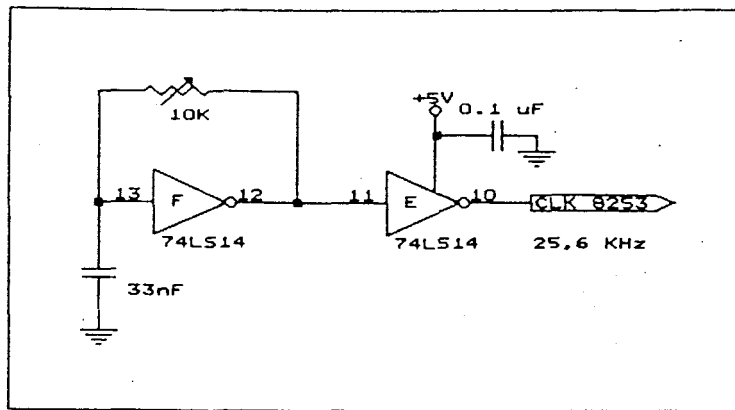
Untuk membuat rangkaian clock dengan frekuensi tersebut digunakan IC Schmitt trigger 74LS14 dengan penambahan komponen R dan C, seperti pada gambar 3.10. Penentuan harga komponen luar ditentukan dengan mengambil harga kapasitor C sebesar 33 nF, dan harga tahanan R ditentukan dengan rumus berikut :

$$f = \frac{1,4}{R \times C} \quad \dots (3-4)$$

Maka :

$$R = \frac{1,4}{f \times C} = \frac{1,4}{25,6 \cdot 10^3 \times 33 \cdot 10^{-9}} = 1657,2 \text{ Ohm}$$

Karena tidak ada harga tahanan yang benar-benar tepat seperti perhitungan maka digunakan tahanan variabel jenis multiturn.



GAMBAR 3.10
RANGKAIAN CLOCK PIT 8253

III.3.2. MODUL PENGINDRA SENSOR

Modul ini merupakan modul pengkondisi sinyal yang berasal dari transduser/sensor menjadi suatu tegangan analog yang akan dikonversikan oleh ADC ke bentuk digital. Pada sistem kontrol yang dibuat, ada 3 rangkaian detektor yaitu detektor temperatur, detektor kelembaban relatif dan detektor intensitas cahaya. Adanya pengkondisian sinyal yang sama dari sensor temperatur dan sensor kelembaban relatif menyebabkan minimisasi rangkaian dapat dilakukan yaitu dengan menambahkan rangkaian multiplexer dan demultiplexer analog.

III.3.2.1. RANGKAIAN MULTIPLEXER DAN DEMULTIPLEXER

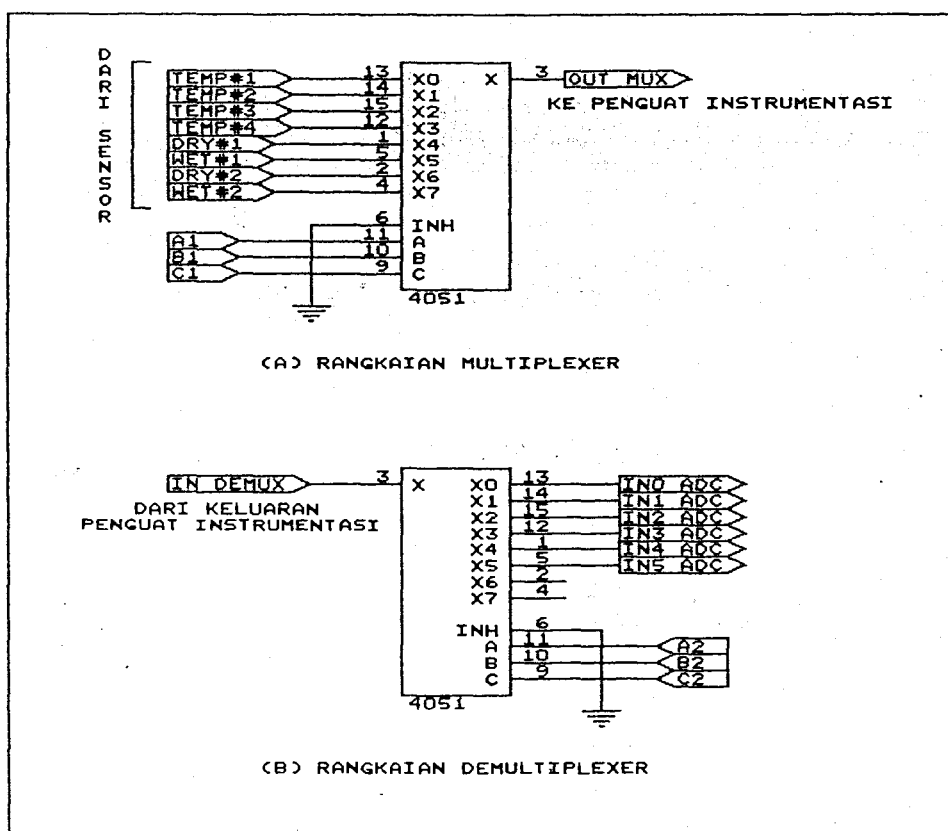
Untuk mendukung proses multiplex dan demultiplex

sinyal analog digunakan IC CMOS 4051 yang merupakan 8 channel analog multiplexer / demultiplexer. Padanya terdapat 3 address input ($A_0 - A_2$), enable input (\bar{E}) yang aktif low, 8 input/output ($Y_0 - Y_7$) dan common input / output (Z). V_{DD} dan V_{SS} merupakan dua pin untuk hubungan tegangan supply ke kontrol masukan digital ($A_0 - A_2, \bar{E}$) input/output ($Y_0 - Y_7, Z$) dapat berada antara V_{DD} sebagai batas positif dan V_{EE} sebagai batas negatif.

Pada rangkaian multiplexer dan demultiplexer yang ditunjukkan pada gambar 3.11, V_{DD} dihubungkan ke +5V, V_{EE} ke -5V dan pin enable (\bar{E}) dihubungkan ke ground. Pin input ($Y_0 - Y_7$) dari rangkaian multiplexer dihubungkan dengan sensor-sensor suhu yang digunakan untuk mengambil besaran temperatur dan kelembaban, sedang pin output (Z) dihubungkan ke rangkaian pengkondisi sinyal, dalam hal ini berupa rangkaian instrumentasi. Pemilihan input yang akan terhubung ke output dilakukan dengan menghubungkan keluaran I/O buffer ($A_1 - C_1$) ke input address ($A_0 - A_2$).

Rangkaian demultiplexer memiliki fungsi untuk menyalurkan keluaran rangkaian pengkondisi sinyal ke channel ADC yang dipakai untuk memproses masukan masing-masing sensor. Pin inputnya (Z) dihubungkan dengan keluaran rangkaian pengkondisi sinyal, sedang pin output ($Y_0 - Y_7$) didistribusikan ke channel ADC ($IN_0 - IN_5$).

Pemilihan input yang akan terhubung ke output dilakukan dengan menghubungkan keluaran buffer I/O ($A_2 - C_2$) ke input address ($A_0 - A_2$).



GAMBAR 3.11

RANGKAIAN MULTIPLEXER DAN DEMULTIPLEXER

III.3.2.2. RANGKAIAN DETEKTOR TEMPERATUR

Pada rangkaian detektor temperatur seperti yang ditunjukkan pada gambar 3.12, digunakan IC sensor LM335 sebagai sensor temperatur. LM335 ini mempunyai tegangan breakdown proporsional terhadap temperatur absolut sebesar 10 mV/°K. Sebagai acuan pengukuran pada 0°C (273°K) maka dibutuhkan tegangan referensi sebesar 2.73 V. Rangkaian penghasil tegangan referensi ini disusun oleh IC dioda referensi presisi 2.5 V teregulasi jenis LM336. Keluaran IC ini kemudian diinputkan ke penguat tegangan berupa non-inverting amplifier dengan menggunakan IC CA3140, untuk mendapatkan tegangan sebesar 2.73 V. Dengan mengambil tahanan umpan balik R_2 sebesar 1K maka diperlukan tahanan R_1 sebesar :

$$\frac{V_o}{V_i} = 1 + \frac{R_2}{R_1} \quad \Rightarrow \quad \frac{2.73}{2.5} = 1 + \frac{1}{R_1}$$

$$\therefore R_1 = 10.87 \text{ K}\Omega$$

Untuk mendapatkan harga R_1 di atas digunakan tahanan jenis multiturn 20K.

Selanjutnya keluaran tegangan referensi dan keluaran sensor diumpankan ke penguat instrumentasi. Penguat ini dioptimalkan untuk mendapatkan impedansi input yang tinggi, *common mode rejection ratio* (CMRR) yang tinggi, penguatan sinyal keluaran dengan noise rendah, serta hanyutan suhu dan waktu yang rendah. Berdasarkan itu untuk penguat instrumentasi ini digunakan IC LF357 karena

memiliki impedansi input yang tinggi dan noise yang rendah. Penguatan pada penguat instrumentasi ini direncanakan sebesar 2 kali. Dengan mengambil harga $R_f = R_6 = 100 \text{ K}\Omega$ dan $R_3 = R_5 = 25 \text{ K}\Omega$ maka harga R_4 dapat dicari dari rumus penguatan berikut :

$$\frac{V_o}{V_1 - V_2} = \frac{R_3 + R_4 + R_5}{R_4} \cdot \frac{R_f}{R_6} \quad \dots (3-5)$$

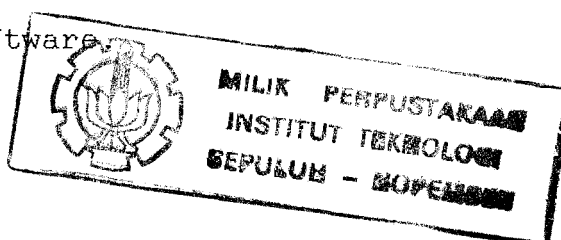
Maka dengan memasukkan harga-harga tahanan yang telah ditentukan, didapat :

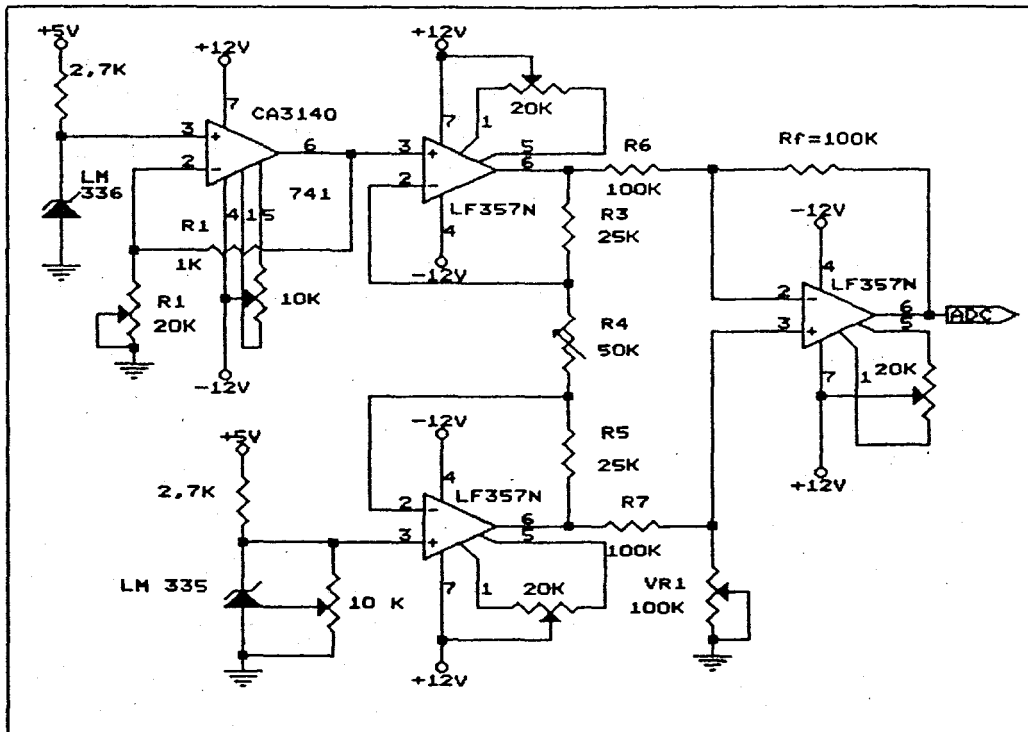
$$\therefore R_4 = 2 \times R_3 = 50 \text{ K}\Omega$$

Sedangkan tahanan R_6 sebesar $100 \text{ K}\Omega$ dapat diubah-ubah untuk menyeimbangkan tegangan common mode.

Pada penguatan sebesar dua kali ini, sinyal menggunakan bagian terbesar dari input range ADC. Hal ini akan menambah kekebalan terhadap noise dan resolusinya. Dengan referensi sebesar 5.12 V , ADC akan mempunyai 20 mV per step. Penguatan sinyal keluaran sensor sebanyak 2 kali akan menyebabkan perubahan output sebesar 20 mV , yaitu sama dengan satu step ADC, untuk tiap kenaikan 1°C . Jadi didapat resolusi sebesar 1°C , dan ini sama dengan akurasi dari sensor.

Untuk meyakinkan distribusi temperatur yang merata di seluruh ruangan, diperlukan 4 buah sensor temperatur. Adapun pemilihan saluran ke rangkaian multiplexer, demultiplexer dan ADC serta perata-rataan keluaran dilakukan dengan software.





GAMBAR 3.12

RANGKAIAN DETEKTOR TEMPERATUR

III.3.2.3. RANGKAIAN DETEKTOR KELEMBABAN RELATIF

Rangkaian detektor kelembaban relatif ini mempunyai bagian pengkondisi sinyal yang sama dengan rangkaian detektor temperatur. Penggunaan metode *psychrometer* untuk mengukur kelembaban relatif, menyebabkan diperlukannya dua buah sensor temperatur yaitu satu disebut *dry bulb* dan satunya lagi disebut *wet bulb*. Alasan digunakannya metode ini adalah karena hasilnya cukup akurat untuk

aplikasi praktis dan memudahkan pengontrolan dari sistem yang direncanakan.

Wet bulb dikondisikan dengan membungkus sensor temperatur dengan kain/sumbu yang mengisap air (porous). Pada penempatannya, dry bulb diletakkan di depan wet bulb dari arah aliran udara yang masuk. Hal ini dimaksudkan untuk menjamin keakuratan pengambilan besaran pada dry bulb, tanpa dipengaruhi efek pendinginan wet bulb.

Rangkaian untuk detektor kelembaban relatif ini diperlihatkan pada gambar 3.12. Untuk meyakinkan kelembaban yang merata di seluruh ruangan, diperlukan 2 buah sensor kelembaban. Adapun pemilihan saluran ke rangkaian multiplexer, demultiplexer dan ADC, perataan keluaran dan konversi ke besaran kelembaban relatif dilakukan dengan software.

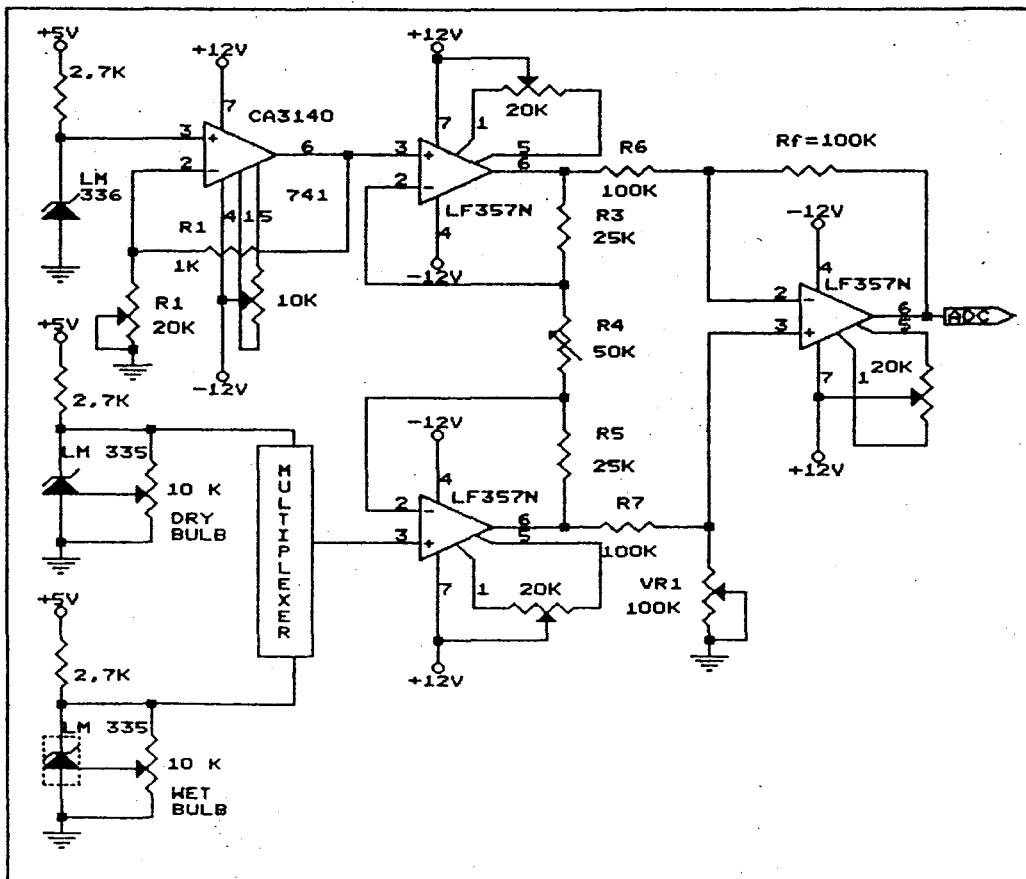
III.3.2.4. RANGKAIAN DETEKTOR INTENSITAS CAHAYA

Pada rangkaian detektor ini, digunakan solar cell sebagai sensor cahaya. Dimensi dari solar cell yang digunakan adalah $5\frac{1}{2}$ cm x 2 cm, dengan tegangan open circuit maksimum sebesar 4.5 V dan arus short circuit maksimum 6 mA.

Pada perencanaan, karena keterbatasan peralatan, rangkaian didesain untuk pengukuran intensitas cahaya hanya sampai 3000 lux. Dari percobaan yang dilakukan,

pada intensitas cahaya sebesar 3000 lux dihasilkan arus short circuit sebesar 540 μA . Penyesuaian dengan referensi ADC membatasi tegangan keluaran hanya sampai 5.12 V. Karena itu tahanan umpan balik pada pengubah arus ke tegangan ditentukan besarnya sebagai berikut :

$$R = \frac{5.12}{540 \cdot 10^{-6}} \approx 9.48 \text{ K}\Omega$$



GAMBAR 3.13

RANGKAIAN DETEKTOR KELEMBABAN RELATIF

Keluaran dari rangkaian pengubah arus ke tegangan ini kemudian dilewatkan ke low pass filter -40 dB/dekade. Filter low pass ini diperlukan untuk menghilangkan noise yang umumnya terjadi pada frekuensi tinggi. Noise ini dapat berasal dari komponen, saluran, transduser, ataupun akibat interferensi. Selain itu filter ini juga dapat menghilangkan spike-spike yang dapat mengganggu sinyal.

Agar sinyal keluaran transduser tidak berubah besarnya maka closed-loop gain filter harus sedekat mungkin dengan 1 di dalam pita lewatnya. Filter yang cocok untuk jenis pemakain ini adalah Butterworth low-pass filter yang merupakan jenis filter yang umum dipakai. Direncanakan frekuensi cut-off untuk filter ini adalah 50 Hz. Dengan mengambil harga $R_1 = R_2 = 10 \text{ K}\Omega$ maka nilai C_1 dapat dihitung sebagai berikut :

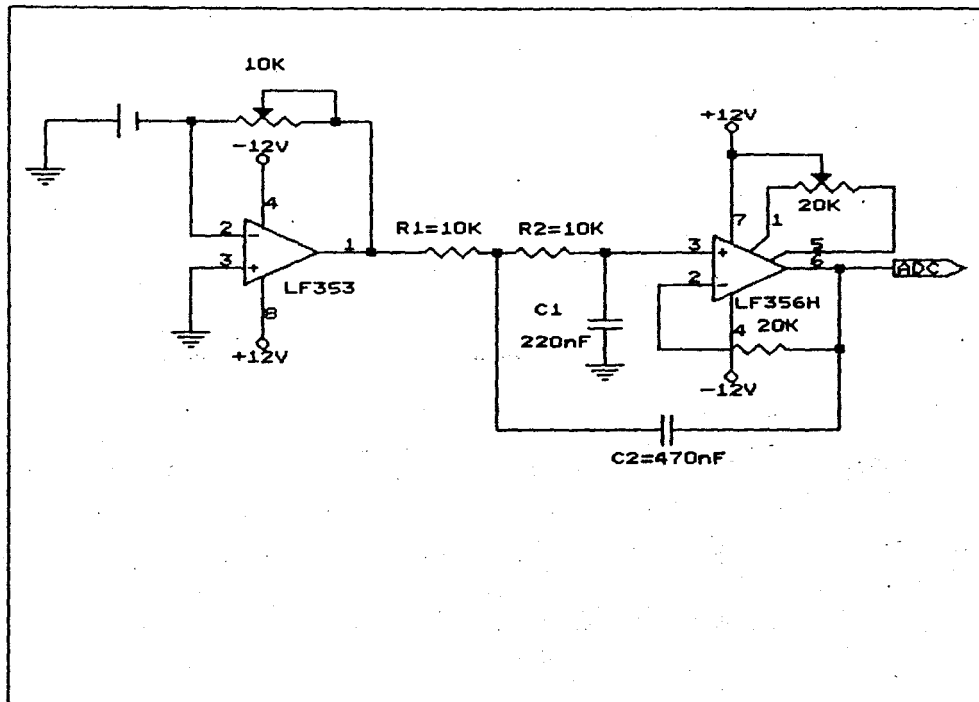
$$C_1 = \frac{0.707}{\omega_c \times R_1} = \frac{0.707}{2.\Pi.50.10.10^3}$$

$$\therefore C_1 = 22 \text{ nF}$$

Harga C_2 ditentukan :

$$C_2 = 2 \times C_1 = 44 \text{ nF}$$

Pendekatan terhadap harga kapasitor yang ada di pasaran menyebabkan harga $C_1 = 220 \text{ nF}$ dan $C_2 = 470 \text{ nF}$. Tahanan R_f digunakan untuk offset dc dan harganya diambil dua kali harga R_1 yaitu sebesar $20 \text{ K}\Omega$.



GAMBAR 3.14

RANGKAIAN DETEKTOR INTENSITAS CAHAYA

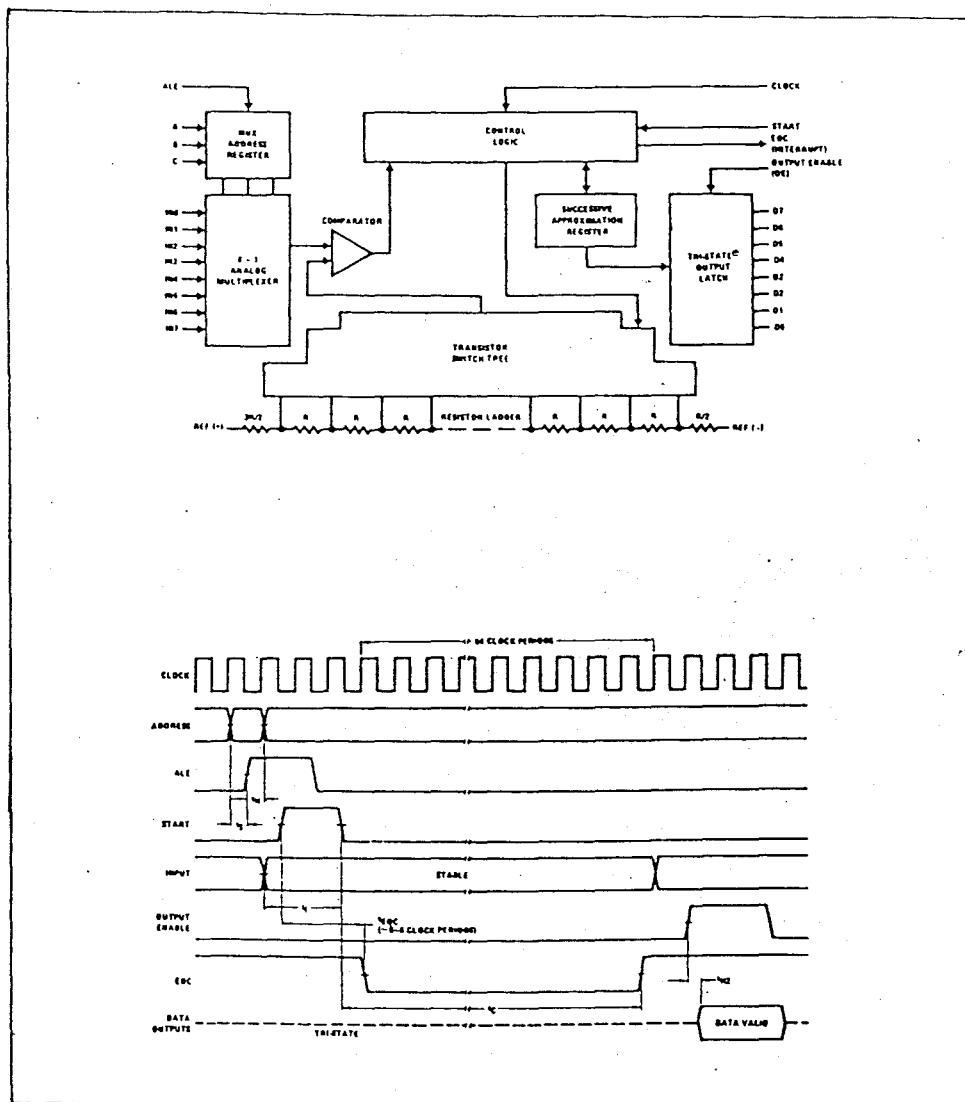
III.3.3. MODUL ANALOG TO DIGITAL CONVERTER

Modul ini merupakan bagian yang mengkonversikan data analog keluaran rangkaian pengindra sensor menjadi data digital. Jenis ADC yang dipakai adalah ADC 0808 buatan National Semiconductor. ADC 0808 ini memiliki resolusi 8 bit dan satu dari ke-8 inputnya dapat disampel dengan memilih 3 channel address. Input analog dari channel yang dipilih dikonversi menjadi data digital 8 bit dengan waktu konversi tipikal 100 μ s. Kemudian keluaran tri-

state latch/buffer yang ada padanya diaktifkan sehingga terhubung langsung ke data bus komputer. ADC 0808 ini memerlukan tegangan referensi external yang menentukan akurasi serta sinyal clock dari 10 sampai 1280 Khz.

ADC 0808 menggunakan teknik *successive approximations* untuk konversi analog ke digital. Successive approximation register, R-2R resistor ladder dan pembanding analog semuanya tercakup pada chip ADC 0808. Dengan referensi tegangan sebesar 5 Volt maka tegangan input dibatasi dari 0 sampai 5 Volt.

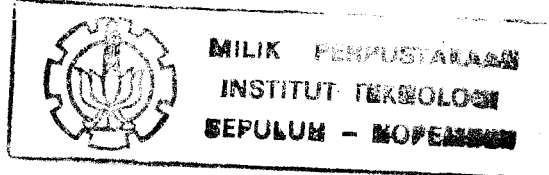
Gambar 3.15 menunjukkan blok diagram dan diagram waktu untuk kontrol sinyal ADC 0808. Proses konversi dimulai dengan mengaktifkan sinyal ALE (Address Latch Enable). Hal ini memuat channel yang dipilih ke multiplexer dan membawa sinyal ke satu dari 8 saluran masukannya ke pembanding. Kemudian dengan mengaktifkan sinyal START, konversi dimulai. Keluaran EOC (End of Conversion) 'low' saat internal successive approximations register mengumpulkan data biner 8 bit. Perubahan dari 'low' ke 'high' dari keluaran EOC mengindikasikan bahwa proses konversi telah selesai dan data dapat dibaca. Dengan memberikan pulsa 'high' ke OE (Output Enable) akan menempatkan data digital hasil konversi ke data bus (yang normalnya dalam keadaan tri-state).

GAMBAR 3.15 ¹⁰

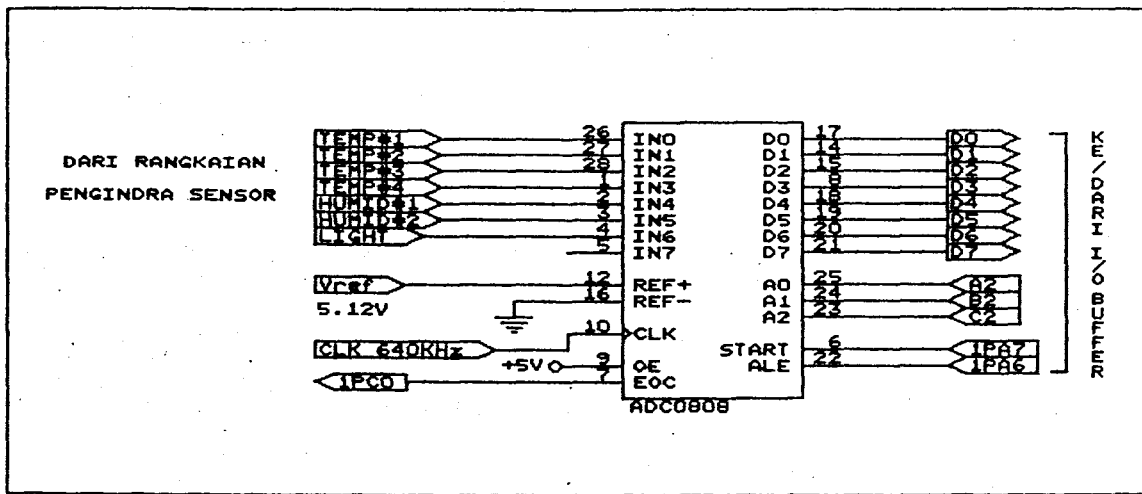
BLOK DIAGRAM DAN DIAGRAM WAKTU ADC 0808

Pada alat yang dibuat, 7 dari 8 saluran masukan ADC 0808 dipakai yaitu 4 untuk rangkaian detektor temperatur (INO-IN3), 2 untuk rangkaian detektor kelembaban relatif

¹⁰ Hogenboom, P., Data Sheet Book 2, Uitgeversmaatschappij Elektuur B.V. (PT.Elex Media Komputindo), Netherlands (Jakarta), 1988, hal.39-40.



(IN4-IN5) dan 1 untuk rangkaian detektor intensitas cahaya (IN6). Adapun pengaturan dan pemilihan channel address dilakukan dengan software.



GAMBAR 3.16
RANGKAIAN ADC

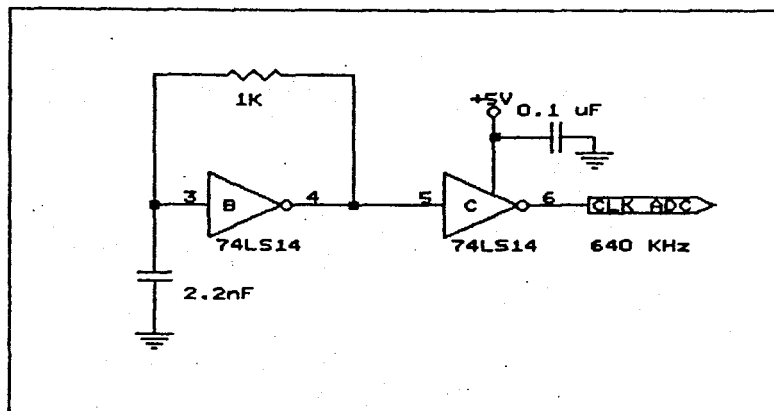
RANGKAIAN CLOCK ADC 0808

Frekuensi clock ADC direncanakan sebesar nilai tipikalnya yaitu 640 KHz. Yang perlu diperhatikan yaitu jika clock ADC 0808 lebih besar dari 500 KHz, diperlukan delay antara sinyal ALE dan START.¹¹ Hal ini dapat dilakukan melalui software ataupun dengan memisahkan pengaktifan ALE dan START.

¹¹ Ibid, hal. 243.

Rangkaian clock ADC ini dibuat dengan menggunakan oscillator RC biasa. Dengan mengambil harga kapasitor sebesar 2.2 nF maka harga tahanan yang diperlukan untuk menghasilkan clock 640 Khz adalah :

$$R = 1.4 / (640 \times 10^3 \times 2.2 \times 10^{-9}) \approx 1 \text{ K}\Omega$$

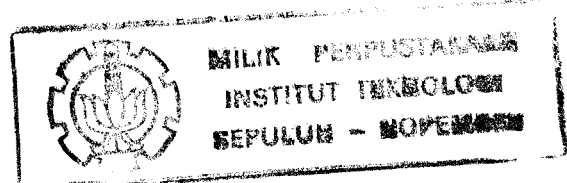


GAMBAR 3.17

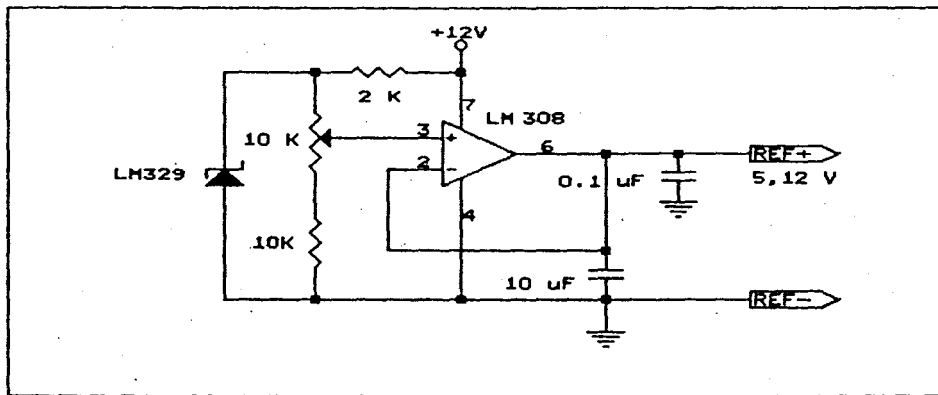
CLOCK ADC 0808

RANGKAIAN PENGHASIL TEGANGAN REFERENSI

Tegangan referensi untuk ADC 0808 ini dibuat sebesar 5.12 Volt. Hal ini dimaksudkan agar ADC memiliki resolusi 20 mV per step karena sampel seluruhnya adalah 256 step. Tegangan sebesar 5.12 V ini diambil dari tegangan diode zener presisi LM 329 yang memiliki keluaran 6.8 V dengan pembagi tegangan. Sebelum diinputkan ke pin Vref(+) ADC



0808, tegangan ini dibuffer dengan LM 308. Kapasitor pemintas 0.1 μF ditambahkan untuk mengatasi 'glitch' yang dapat menyebabkan ketidakakuratan proses konversi.



GAMBAR 3.18

RANGKAIAN PENGHASIL TEGANGAN REFERENSI

III.3.4. MODUL DRIVER

Sistem kontrol yang direncanakan adalah close-loop sistem. Dalam hal ini besaran parameter hasil akuisisi data dalam sistem dibandingkan dengan suatu set point yang dapat diinputkan. Bila keduanya tidak sama maka ada kontrol yang dilakukan sebagai umpan balik ke sistem. Modul driver ini memegang peranan sebagai peralatan yang mendukung kontrol umpan balik tersebut disamping sebagai fasilitas pelengkap sistem. Jenis kontrol yang dilakukan

ada dua macam yaitu kontrol diskontinyu (ON-OFF) dan kontinyu (PID controller), sedangkan alur proses kontrolnya diatur dengan perangkat lunak.

III.3.4.1. RANGKAIAN PENGAKTIF POMPA PENGISAP

Ada 2 rangkaian pengaktif pompa pengisap yang dibuat yaitu satu untuk fasilitas penyiraman/pemupukan dan satu lagi sebagai pengaktif sprayer. Sprayer ini berfungsi sebagai pengatur kelembaban relatif dalam sistem. Bila harga kelembaban relatif lebih kecil dibanding set point, maka sprayer akan menyembrotkan air yang akan ditiup oleh fan sehingga menjadi butir-butir halus. Cara ini merupakan cara pengaturan kelembaban yang dipakai. Sedangkan jenis kontrol rangkaian ini bersifat diskontinyu (on-off).

Rangkaian pengaktif pompa pengisap ditunjukkan pada gambar 3.19. Pompa pengisap itu sendiri merupakan motor +12V DC yang akan diaktifkan dengan kontak relay. Relay yang digunakan adalah relay SPDT dengan perlawanan 400 Ω . Dengan perlawanan sebesar itu maka diperlukan arus sebesar 30 mA untuk mendrive relay tersebut. Untuk itu dipakai transistor buffer BC140 sebagai driver relay. Transistor ini berdasarkan spesifikasinya memiliki h_{fe} sebesar 40-120 dan $I_{cmax} = 1A$. Pengukuran memberikan harga h_{fe} sebesar 80, sehingga untuk mendrive relay

diperlukan arus basis sebesar :

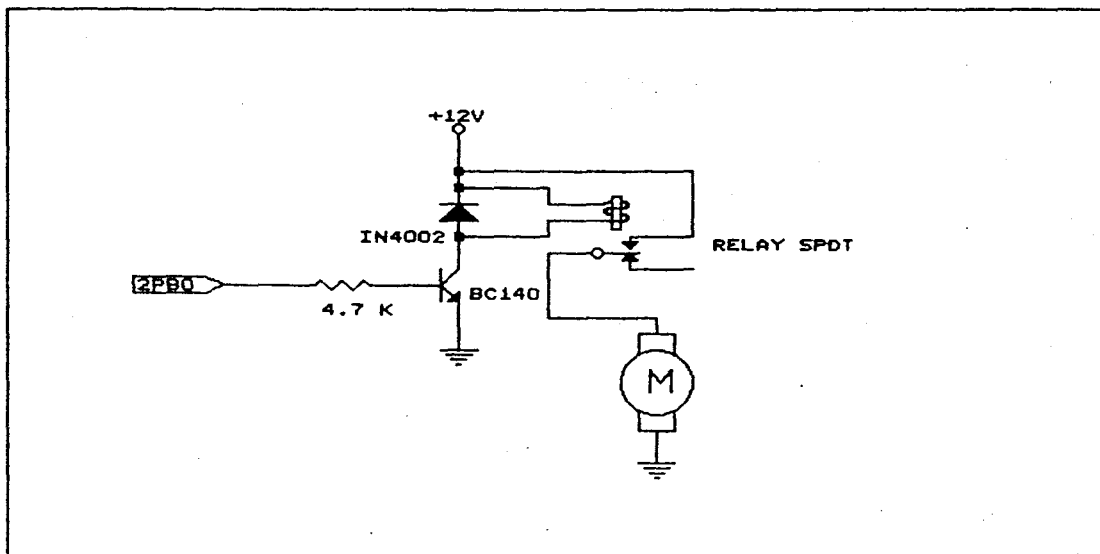
$$I_b = I_c / h_{fe} = 30 / 80 \text{ mA} = 0,375 \text{ mA}.$$

Dengan level high tegangan output port 8255 ke peripheral minimumnya adalah 2,4V maka pada saat transistor ON agar arus yang mengalir pada basis sebesar 0,375 mA, pada basis diberi tahanan pembatas arus R_b sebesar :

$$R_b = (2,4 - 0,7) / 0,375 = 4,53 \text{ K}\Omega$$

Diambil harga $R_b = 4,7 \text{ K}\Omega$.

Jadi keluaran 'high' dari port PPI 8255 akan menukar kontak relay ke tegangan +12 V sehingga pompa aktif.



GAMBAR 3.19

RANGKAIAN PENGAKTIF POMPA PENGISAP

Dioda yang dipasang reverse pada kumparan relay berguna untuk melindungi transistor, terutama saat transistor off. Sifat beban induktif, seperti relay, adalah mempertahankan arus untuk beberapa saat setelah off. Karenanya dapat membangkitkan imbasan tegangan sepanjang induktor yang biasanya cukup besar untuk mem-break down transistor bila dioda tidak dipasang. Bila kumparan relay konduksi atau transistor on, dioda akan tercatu balik sehingga tidak konduksi. Pada saat tegangan imbas mencapai 0,7 V dioda akan konduksi dan mensuplai suatu lintasan balik untuk arus imbas. Tegangan yang melintasi kumparan tetap dipertahankan pada 0.7 V sehingga mengamankan transistor.

III.3.4.2. RANGKAIAN PENGGERAK MOTOR STEPPER

Motor stepper ini digunakan untuk menarik peneduh khusus yang biasa digunakan untuk mengurangi intensitas cahaya matahari sampai beberapa puluh persen. Rangkaian ini digunakan untuk pengaturan intensitas cahaya matahari yang jatuh ke model rumah kaca yang dibuat.

Motor stepper yang digunakan mempunyai rating tegangan 12 V/phase, arus 0,6 A/phase dan putaran $1,8^\circ/\text{step}$. Untuk menggerakkan motor stepper ini diperlukan suatu rangkaian driver. Rangkaian ini diimplementasikan dengan transistor-transistor switching seperti pada

gambar 3.20.

TIP 120 merupakan transistor darlington yang berdasarkan spesifikasinya memiliki $h_{fe} = 1000$ dan $I_{cmax} = 8$ Ampere. Guna memenuhi rating arus stepper tersebut maka arus basis yang diperlukan adalah $I_b = I_c/h_{fe} = 0,6/1000$ A = 0,6 mA. Keluaran peripheral port PPI 8255 hanya dapat mensuplai arus sebesar 0,2 mA, karena itu keluaran port PPI 8255 dibuffer dulu agar arusnya cukup kuat. Untuk ini digunakan IC 74LS06 (inverter dengan kolektor terbuka) yang mempunyai batas tegangan maksimum 30 Volt dan arus keluaran level rendah sebesar 40 mA. Agar transistor on saat didrive 'low' maka basis diberi tahanan pembatas arus R_b sebesar :

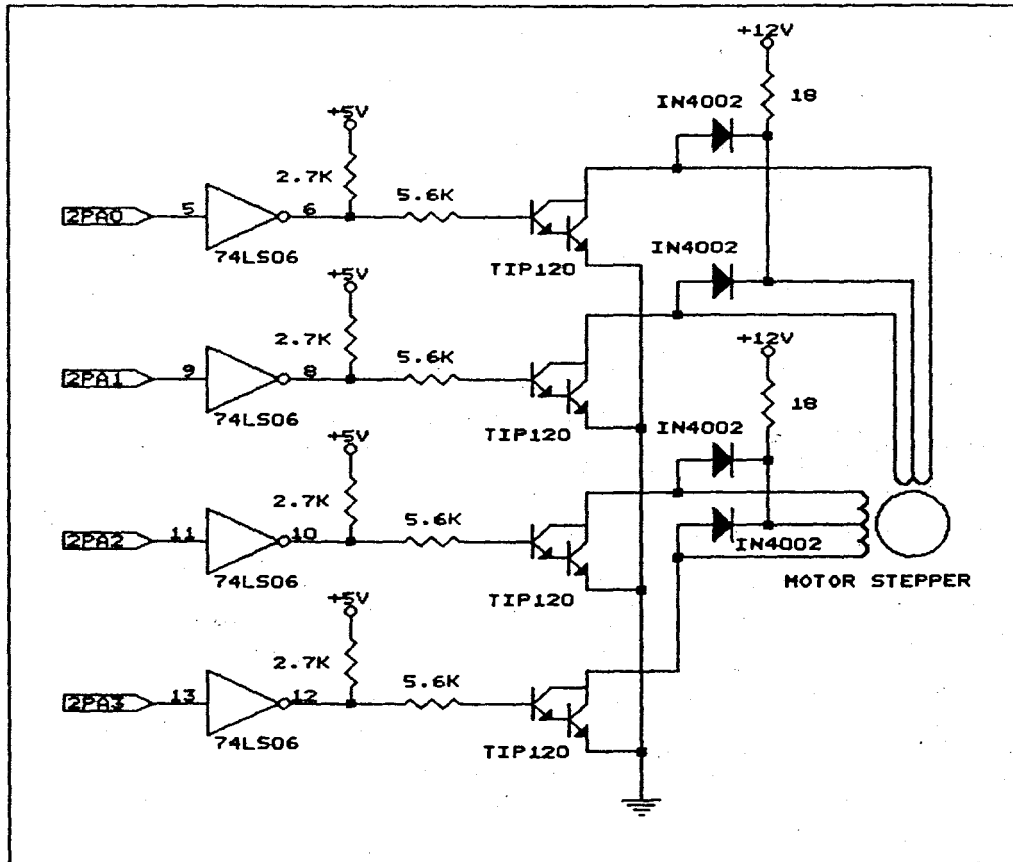
$$R_b = (V_{cc} - V_{be})/I_b = (5 - 1,4)/0,6 = 6 \text{ K}\Omega$$

Diambil harga $R_b = 5,6 \text{ K}\Omega$.

Tegangan supply untuk motor stepper ini bukanlahlah level tegangan TTL dan daya yang diserap cukup besar, karena itu dibuat power supply sendiri dengan $V_{cc} = +12$ V dan arus maksimum 2 A. Dan untuk memastikan bahwa arus yang mengalir dari V_{cc} sebesar 0,6 A maka dipasang tahanan pembatas arus sebesar :

$$\begin{aligned} R_1 = R_2 &= (V_{cc} - V_{ce})/I_c \\ &= (12 - 0,7)/0,6 \\ &= 18,8 \Omega \end{aligned}$$

Diambil harga $R_1 = R_2 = 18 \Omega/10 \text{ W}$.



GAMBAR 3.20

RANGKAIAN DRIVER MOTOR STEPPER

III.3.4.3. RANGKAIAN PENGGERAK FAN

Fan, dalam sistem yang direncanakan, mempunyai beberapa fungsi yaitu sebagai peniup butiran air dari sprayer untuk pengaturan kelembaban, sebagai penyuplai angin yang diperlukan oleh tanaman dan sebagai pemerata panas dari pemanas ke model rumah kaca yang dibuat.

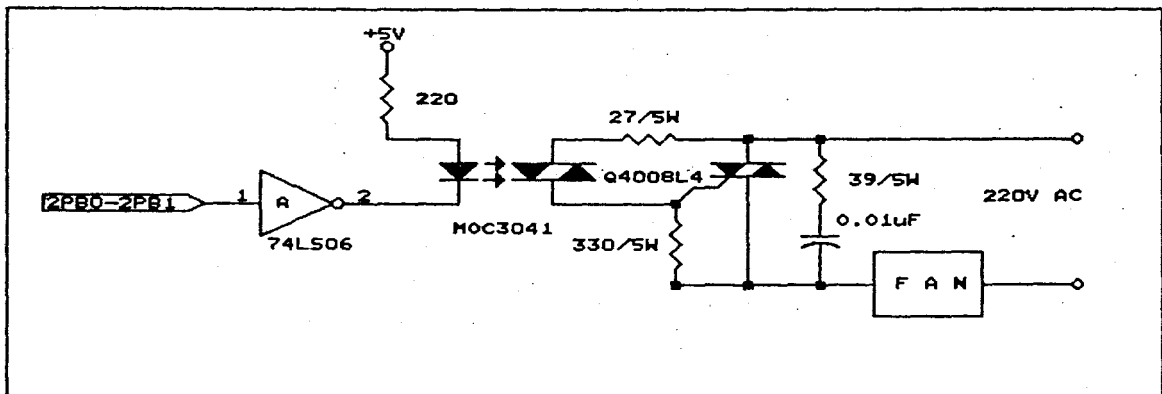
Rangkaian penggerak fan ini merupakan *Isolated hybrid solid-state relay* yang diimplementasikan dengan optocoupler dan triac sebagai switch. Optocoupler yang digunakan adalah jenis triac driver output MOC 3041 yang memiliki rangkaian zero voltage crossing didalamnya. Dan jenis triac yang digunakan adalah triac Q4008L4 yang memiliki tegangan breakover ($V_{\text{BRO}} = 400 \text{ V}$) dan $I_{\text{T(rms)}} = 8 \text{ A}$. Kedua komponen di atas dipakai untuk driver rangkaian dengan tegangan puncak jala-jala sebesar 220 V.

Berdasar spesifikasinya, MOC 3041 memerlukan arus trigger output $I_{\text{F}} = 15 \text{ mA}$ sampai maksimum $I_{\text{FT}} = 60 \text{ mA}$ untuk me-latch output, sedang tegangan forward (V_{F}) saat itu tipikalnya sebesar 1,3 V dan maksimum 1,5 V. Sedang triac Q4008L4 memiliki I_{GT} maksimum sebesar 25 mA, V_{GT} minimum 0,2 V dan maksimum 2,5 V, serta I_{OH} maksimum sebesar 50 mA.

Rangkaian penggerak fan ini ditunjukkan pada gambar 3.21. Pin 1 dari MOC 3041 dihubungkan dengan Vcc sebesar 5 V dan pin 2 dihubungkan dengan keluaran PPI 8255 yang telah dibuffer dengan 74LS06. Keluaran high dari PPI akan mengaktifkan rangkaian karena saat ini fan terhubung dengan tegangan jala-jala.

Tahanan 220 Ω antara Vcc dengan pin 1 diperhitungkan untuk memberikan arus trigger sebesar 15 mA, sesuai dengan rating MOC 3041. Tahanan 39 Ω dan kapasitor 0,01

μF digunakan sebagai rangkaian snubber triac. Rangkaian snubber ini dimaksudkan untuk menghilangkan pengaruh transient dari saluran dan juga pengaruh dari beban induktif yang dapat mengaktifkan triac akibat terlampaunya rating static dv/dt dari triac. Penentuan rangkaian snubber ini memperhitungkan power faktor dari beban, karenanya bersifat empiris.



GAMBAR 3.21

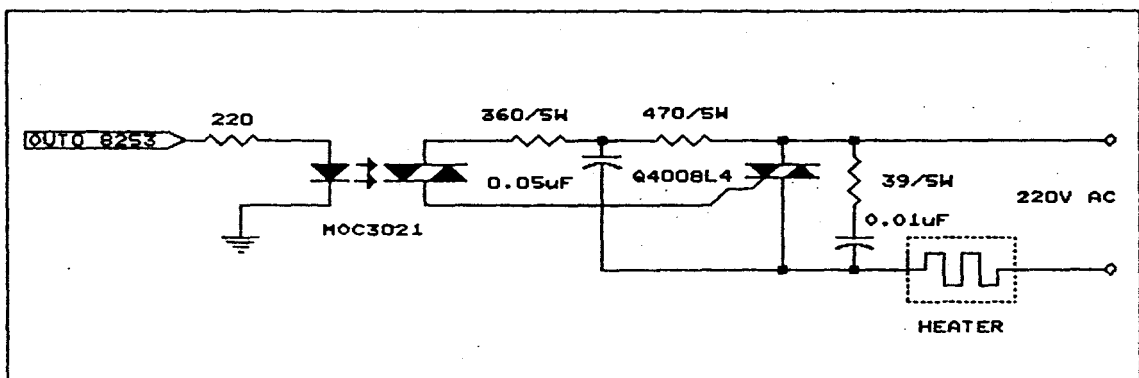
RANGKAIAN PENGGERAK FAN

III.3.4.4. RANGKAIAN PENGATUR PEMANAS

Rangkaian pengatur pemanas ini pada dasarnya sama dengan rangkaian penggerak fan yaitu menggunakan rangkaian solid state power switching. Perbedaannya yaitu digunakannya metoda phase control, tidak lagi zero voltage switching. Bentuk gelombang dan daya efektif yang

diberikan ke beban dibentuk oleh pewaktuan pulsa trigger yang diberikan ke gate triac.

Gambar 3.22 memperlihatkan rangkaian pengatur pemanas yang dibuat. Triac yang digunakan adalah tipe Q4008L4. Sedang jenis optoisolator yang digunakan adalah MOC 3021 yaitu triac driver tanpa zero voltage switching. Rating dari MOC 3021 ini yaitu : I_F tipikal = 8 mA dan maksimum 15 mA dengan I_{Fmax} = 60 mA, sedang V_F tipikalnya sebesar 1,15 V dan maksimum 1,5 V.



GAMBAR 3.22

RANGKAIAN PENGATUR PEMANAS

Pewaktuan pulsa trigger dikontrol dengan keluaran dari PIT 8253. Keluaran high dari PIT ini akan memberikan arus trigger sebesar 15 mA ke input LED dari MOC 3021 dengan memasang tahanan sebesar 220 Ω . Dengan demikian

triac akan aktif dan menghidupkan pemanas. Kontrol kontinyu dilakukan dengan input jumlah sampling yang diberikan ke PIT 8253. Jumlah sampling ini akan mengatur conduction angle dari triac, dimana makin besar conduction angle ini maka makin besar arus yang diberikan ke beban. Dengan sampling = 255 maka tidak ada arus ke beban sedang bila sampling = 0 maka arus mengalir secara penuh ke beban. Dengan makin besar arus yang mengalir ke beban, dalam hal ini pemanas, maka makin besar panas yang dihasilkan. Untuk jelasnya, pada gambar 3.23 ditunjukkan siklus penyalan triac seperti yang dijelaskan di atas.

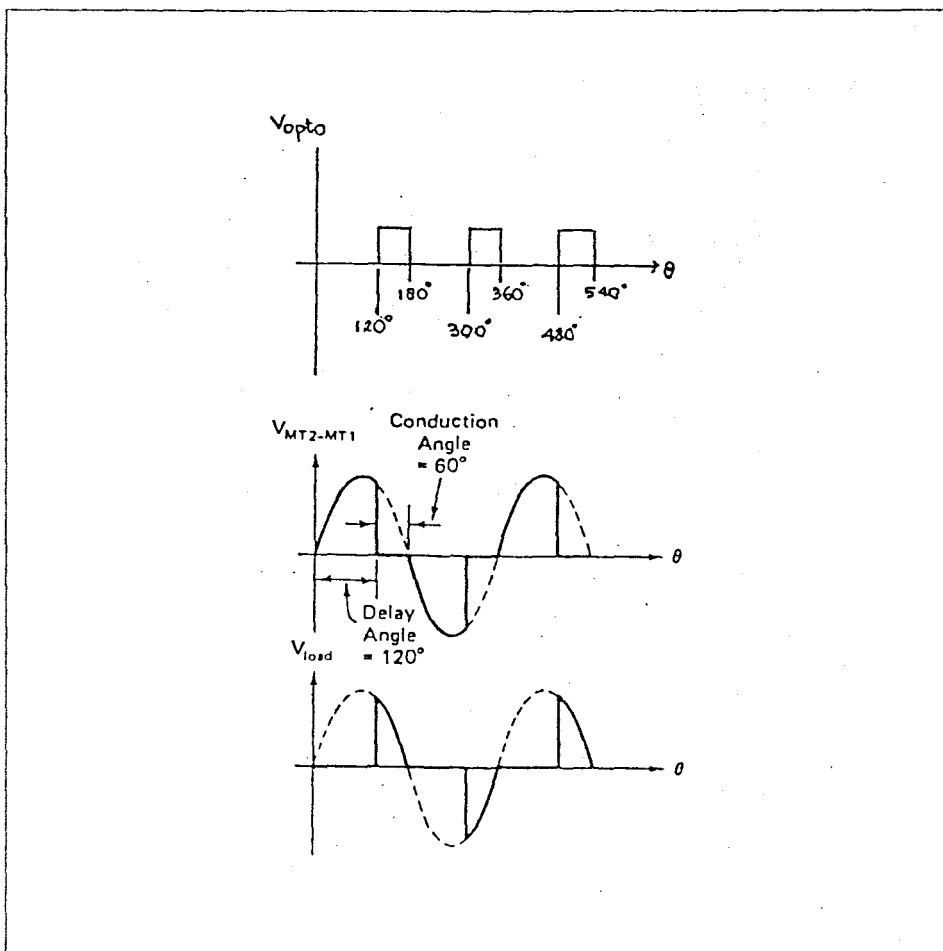
III.3.4.5. RANGKAIAN DIMMER LAMPU FLOURESCENT

Lampu flourescent atau yang lebih dikenal dengan sebutan lampu TL sering digunakan untuk pencahayaan tanaman anggrek. Untuk pengaturan pencahayaan maka direncanakan suatu rangkaian yang dapat mengatur besar intensitas cahaya dari lampu flourescent tersebut.

Karakteristik dari lampu ini adalah bahwa cahaya yang dihasilkan dikontrol oleh perubahan arus dan bukan dengan mengubah tegangannya. Ini disebabkan karena lampu jenis ini menghasilkan potensial ionisasi yang konstan melintasi terminal-terminalnya. ¹²

¹² Irving M. Gottlieb, *Power Control with Solid State Devices*, Reston Publishing Company, Inc., 1985, hal.273.

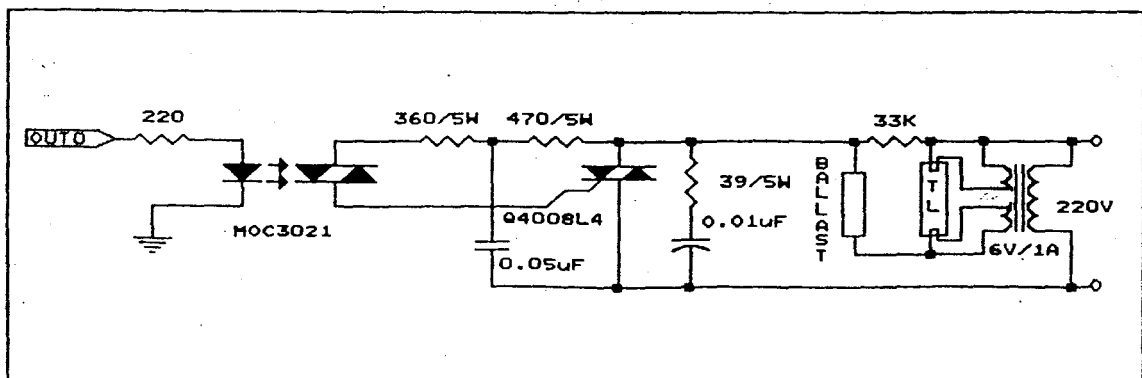
Perubahan arus akan terjadi apabila impedansi dari ballast yang dipasang seri dengan lampu tersebut dibuat tetap. Hal ini dikarenakan ballast, yang merupakan induktor berimpedansi tinggi, menyerap perbedaan tegangan puncak yang diberikan dengan tegangan yang diperlukan untuk mempertahankan ionisasi.



GAMBAR 3.23

SIKLUS PENYALAN TRIAC

Pengubahan harga rata-rata dari arus lampu untuk tujuan pengaturan besarnya intensitas cahaya yang dihasilkan, dimungkinkan dengan penggunaan kontrol phase dari tegangan jala-jala. Namun hal lain yang perlu diperhatikan dalam rangkaian kontrol ini adalah jenis kontrol phase yang digunakan. Jika digunakan kontrol phase setengah gelombang, komponen dc dari pulsa akan menyebabkan saturasi pada inti besi transformer ballast. Akibatnya impedansi akan turun dan dapat mengganggu pengontrolan arus lampu. Oleh karena itu diperlukan kontrol phase gelombang penuh yang tidak memiliki komponen dc. Berdasar ketentuan tersebut, rangkaian dimmer ini dibuat dengan menggunakan rangkaian solid state power switching dengan metoda phase control, seperti ditunjukkan pada gambar 3.24. Adapun intensitas maksimum yang dapat dikontrol adalah kira-kira 1350 lux.

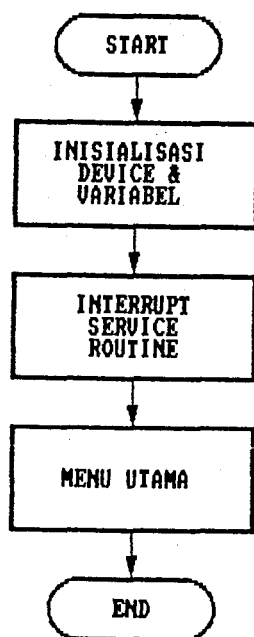


GAMBAR 3.24

RANGKAIAN DIMMER LAMPU FLOURESCENT

III.4. PERENCANAAN SOFTWARE

Perangkat lunak yang dibuat untuk menunjang kerja perangkat keras dibuat dengan menggunakan Turbo C++ versi 1.0 keluaran Borland. Program yang dibuat ini dilengkapi dengan fasilitas tampilan grafik dan pustaka untuk memudahkan pengguna untuk melihat respon sistem dan mendapatkan informasi tentang beberapa jenis tanaman anggrek budidaya. Gambaran umum perangkat lunak yang dibuat ditunjukkan dengan diagram alir pada gambar 3.25.



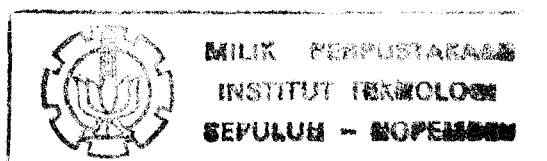
GAMBAR 3.25

DIAGRAM ALIR PERANGKAT LUNAK

Pada menu utama terdapat beberapa pelayanan yang diberikan yaitu setting, monitoring, control dan pustaka. Pada setting, dilakukan penentuan setpoint dari parameter-parameter yang akan dikontrol serta pengkondisian peralatan yaitu sistem penyiraman/pemupukan (waktu, tanggal, kerap dan lama penyiraman/pemupukan) dan fan (on/off). Monitoring menunjukkan kondisi peralatan dan data hasil pengambilan parameter, baik dalam bentuk angka-angka maupun tampilan secara grafis. Pilihan control berguna mengatur pengaktifan dan penghentian sistem kontrol serta reset peralatan. Sedangkan pilihan pustaka menyediakan beberapa file teks berisikan informasi mengenai kebutuhan primer dari beberapa anggrek budidaya yang dapat ditampilkan pada monitor.

III.4.1. INISIALISASI PORT DAN TIMER

Inisialisasi ini pertama kali dilakukan saat software dijalankan yaitu untuk menentukan mode kerja yang dipakai oleh PPI 8255 dan PIT 8253. Kedua 8255 yang dipakai diprogram untuk bekerja sebagai mode 0 dan 8253 bekerja sebagai mode 1. Pada 8255#1, port A dan port Cupper digunakan sebagai port output sedang port B dan port Clower digunakan sebagai port input. Pada 8255#2, semua port dipakai sebagai port output. Inisialisasi ini dinyatakan sebagai berikut :



```

void InitDevice(void)
{
    outportb(CW1,0x83);
    outportb(CW2,0x80);
    outportb(CWCTR,0x12);
    outportb(CTR0,0xff);
    outportb(CWCTR,0x52);
    outportb(CTR1,0xff);
}

```

III.4.2. INTERRUPT SERVICE ROUTINE

Routine ini memegang peranan penting dalam hal pengontrolan sistem, pengambilan data parameter, pengaktifan peralatan dan pengaturan kerja perangkat keras secara keseluruhan. Routine ini pada dasarnya merupakan clock tick interrupt yang memanfaatkan interrupt software pada komputer yaitu INT 08H.

Alur kontrol dari routine ini ditunjukkan pada gambar 3.26. Routine ini disisipkan pada bagian awal program pemroses interupsi timer pada BIOS sehingga routine yang dibuat menjadi bagian yang juga dilakukan setiap terjadi perubahan detak clock. Pertama yang harus dilakukan adalah mengutip vektor address asli dari interupsi timer kemudian menggantikannya dengan address dari routine handler yang dibuat. Pada akhir pelayanan program, handler yang lama direset. Implementasi dalam bahasa pemrograman yang dipakai adalah sebagai berikut :

```

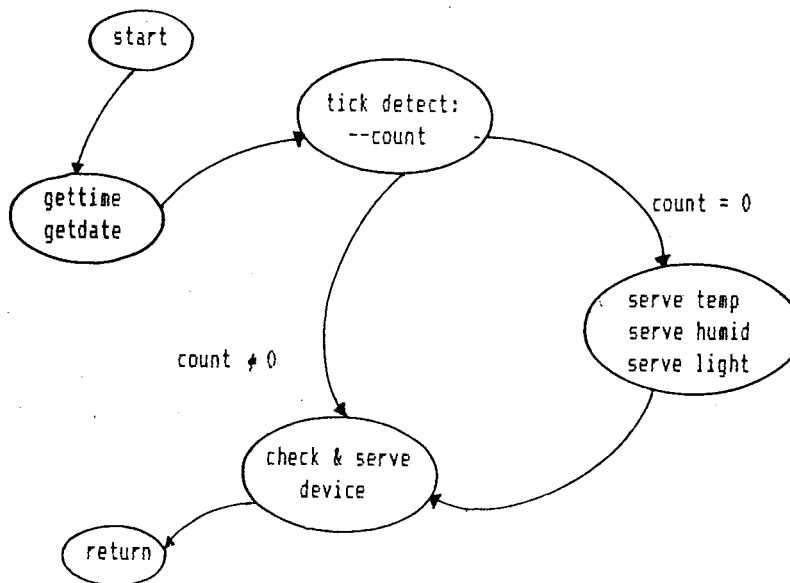
void interrupt (*oldhandler)();

```

```

main()
{
    oldhandler = getvect(0x08);
    setvect(0x08, ISR);
    ....
    setvect(0x08, oldhandler);
    return 0;
}

```

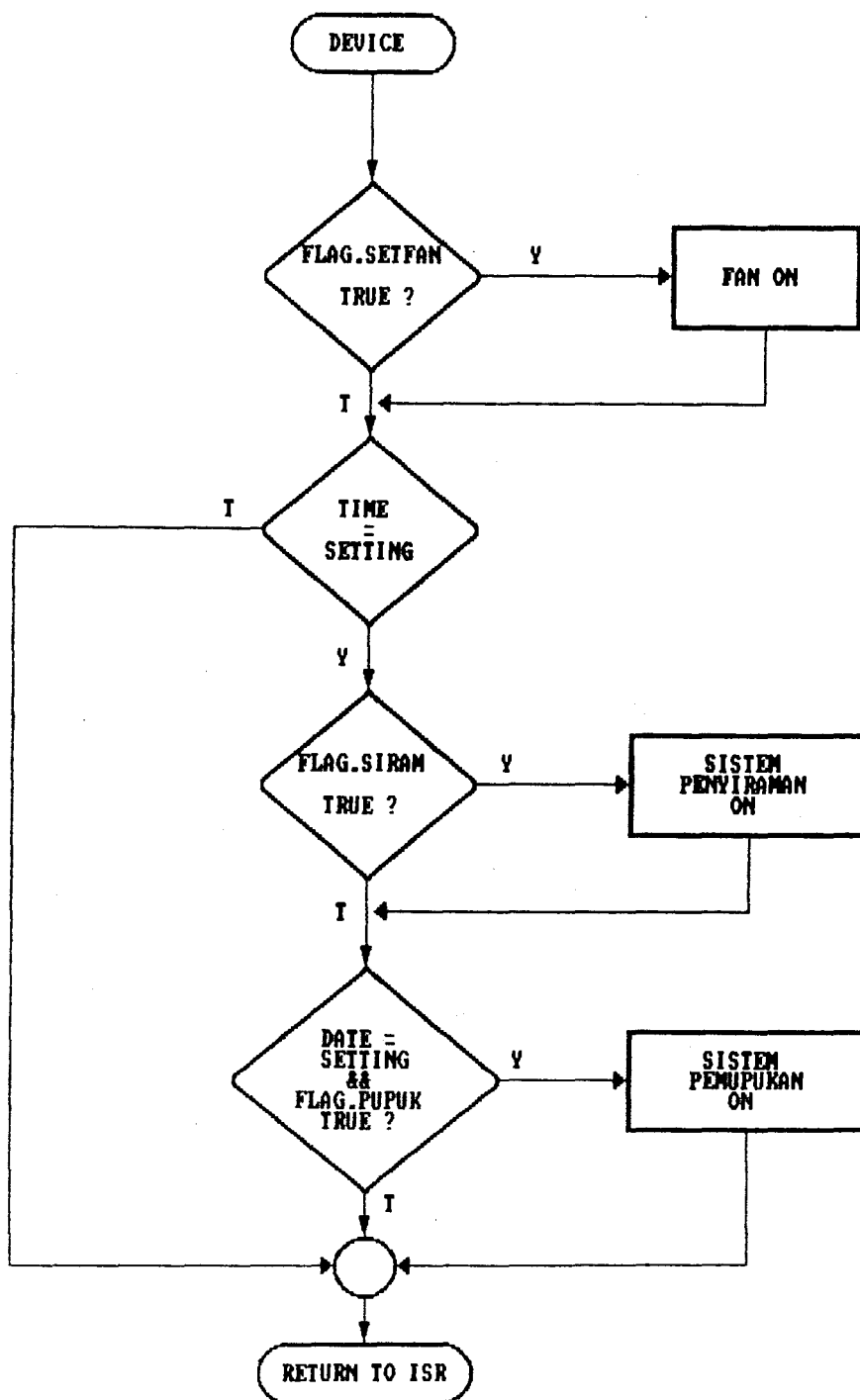


GAMBAR 3.26

ALUR KONTROL INTERRUPT SERVICE ROUTINE

III.4.2.1. PELAYANAN SETTING PERALATAN

Pelayanan ini terus-menerus dilakukan sejalan dengan perubahan tick dari timer. Pengecekan dilakukan dengan membandingkan setting waktu dan tanggal dengan waktu dan tanggal real time komputer. Bila sama, dengan setting peralatan 'on', maka peralatan diaktifkan. Diagram alir pelayanan peralatan ini diberikan pada gambar 3.27.



GAMBAR 3.27

DIAGRAM ALIR PELAYANAN PERALATAN

III.4.2.2. KONTROL UMPAN BALIK

Proses kontrol umpan balik yang dipakai berupa kontroler on-off dan kontroler PID. Pada kontroler on-off, peralatan diaktifkan dan dinonaktifkan dengan memberikan sinyal enable dan disable ke driver rangkaian, misalnya untuk kontrol fan dan sprayer.

Kontroler PID digunakan untuk kontrol kontinyu yang mengatur besarnya nilai count yang diberikan ke PIT 8253 untuk pengaturan sudut penyalan triac sehingga didapatkan suatu phase kontrol. Kontroler ini dipakai untuk pengontrolan pemanas dan dimmer lampu flourescent.

Persamaan untuk PID kontroler ini telah ditunjukkan pada persamaan 2-16. Terlihat bahwa persamaan tersebut memerlukan harga awal yaitu untuk $E_p[n-1]$ dan $\Delta E_p[n-1]$. Dalam aplikasi, harga awal untuk kedua variabel tersebut diset = 0.

Nilai dari konstanta K_i , K_p dan K_d untuk kontroler PID ditentukan dengan cara mencoba-coba sehingga didapatkan respon sistem yang cukup baik. Dalam artian ini, respon secara cepat mencapai kondisi yang diharapkan dan dapat stabil pada kondisi itu. Sedangkan harga T bersesuaian dengan setting interval sampling yang diinputkan.

Adapun implementasi persamaan PID kontroler dalam bahasa pemrograman yang dipakai adalah sebagai berikut :

```

int PID(void)
{
    int VarT;

    preverr = t_err;
    err = set point - sample;
    t_int += Ki * t_err * tsp;
    H = Kp*t_err + t_int + Kp * Kd *(t_err-preverr)/tsp;
    return(VarT);
}

```

III.4.2.3. PELAYANAN SAMPLING PARAMETER

Pengambilan atau sampling terhadap besaran-besaran yang diukur seperti temperatur, kelembaban relatif dan intensitas cahaya serta aksi kontrol umpan balik ke sistem direncanakan secara sekuensial dan diatur tiap interval waktu tertentu.

Interval sampling ditentukan sendiri oleh pengguna dengan cara memasukkan lama waktu sampling (dalam satuan detik) ke fasilitas setting time pada software pendukung. Bila tidak ada pemasukan data, maka waktu sampling diset selama 30 detik. Dalam hitungan timer tick maka waktu sampling yang diset dikalikan dengan 18.2, yaitu banyaknya interupsi periodik yang terjadi setiap detik. Setiap terjadi interupsi timer hitungan tersebut akan dikurangi 1. Bila hitungan mencapai nol maka pelayanan dan pengambilan data parameter dilakukan dan nilai hitungan diperbaharui lagi seperti semula.

Proses sampling parameter melalui ADC dilakukan dengan prosedur berikut :

```

unsigned char GetSample(unsigned char channel)
{
    unsigned char eoc=0,data;

    outportb(PA1,channel ; 0x40);    /* sinyal ALE */
    outportb(PA1,channel ; 0xc0);    /* sinyal START */
    outportb(PA1,channel);
    while(eoc) eoc = inportb(PC1) & 1;
    while(!eoc) eoc = inportb(PC1) & 1;
    data = inportb(PB1);
    return(data);
}

```

Passing parameter *channel* digunakan untuk memilih saluran ADC, multiplex dan demultiplex. Seperti yang telah dijelaskan sebelumnya, pemilihan ini dikontrol oleh keluaran I/O buffer yaitu A₁ - C₁ dan A₂ - C₂. Format logika pemilihan channel ini ditabelkan pada tabel 3-2.

TABEL 3-2

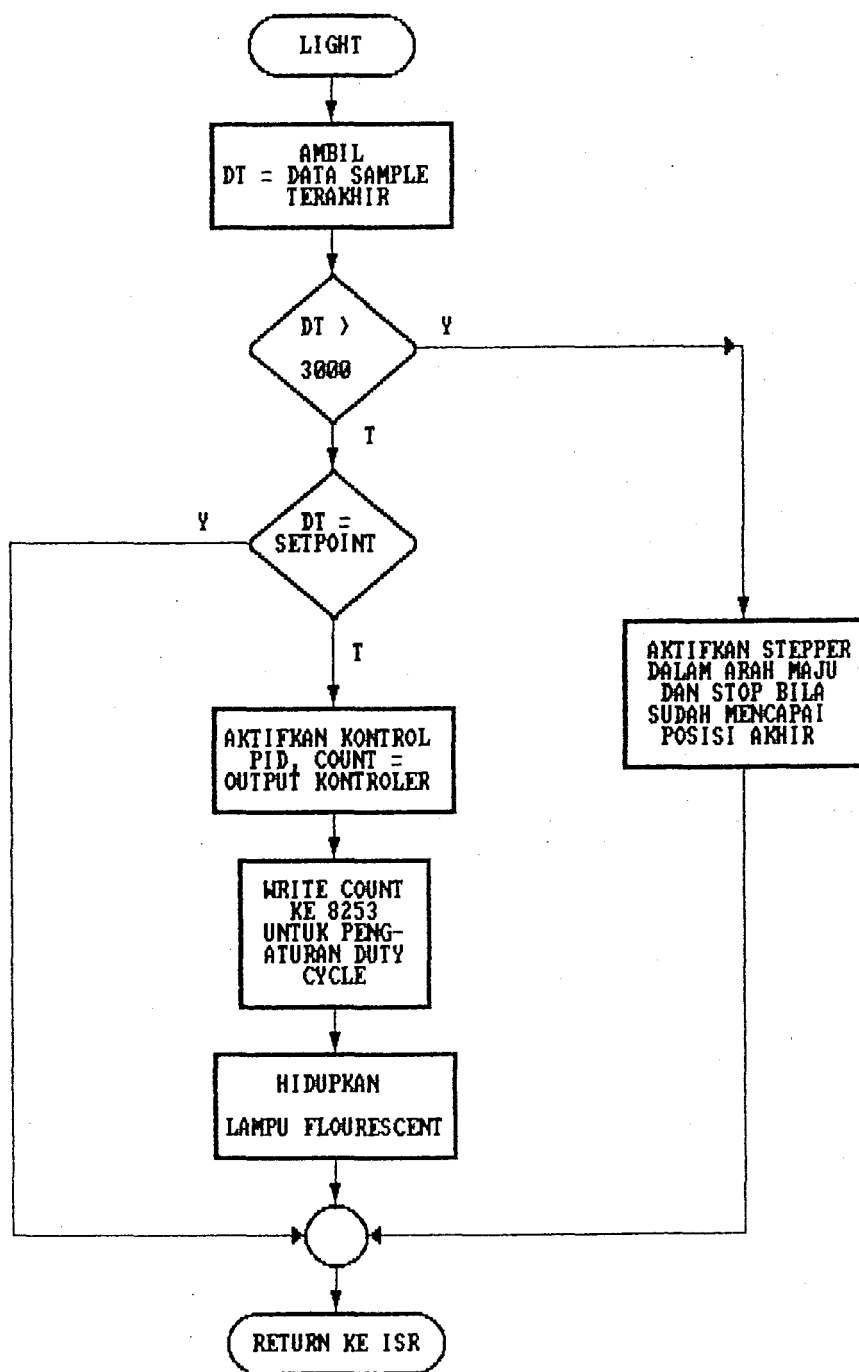
FORMAT PEMILIHAN CHANNEL SAMPLING PARAMETER

PARAMETER	C1	B1	A1	C2	B2	A2
TEMP#1	0	0	0	0	0	0
TEMP#2	0	0	1	0	0	1
TEMP#3	0	1	0	0	1	0
TEMP#4	0	1	1	0	1	1
DRY#1	1	0	0	1	0	0
WET#1	1	0	1	1	0	0
DRY#2	1	1	0	1	0	1
WET#2	1	1	1	1	0	1
LIGHT	X	X	X	1	1	0

Pada awal program dijalankan, dilakukan sampling untuk mendapatkan kondisi awal dalam model rumah kaca. Hasil sample ini dijadikan harga awal setting parameter. Selanjutnya operator dapat memasukkan setting parameter yang ingin dikondisikan.

Setelah dilakukan pengambilan sample parameter-parameter yang diukur maka dilakukan perbandingan antara hasil sample dengan setting yang dilakukan. Bila tidak dalam range yang ditentukan maka dilakukan kontrol umpan balik ke sistem.

Bila hasil sample besaran intensitas cahaya tidak sesuai dengan set point, maka dilakukan pengaturan intensitas dari lampu flourescent. Besarnya intensitas cahaya yang dipancarkan diatur untuk memenuhi range setting. Kontroler yang dipakai disini adalah PID kontroler. Keluaran kontroler ini diinputkan ke 8253 untuk mendapatkan phase kontrol yang digunakan untuk pengubahan harga rata-rata arus lampu sebagaimana yang telah diuraikan pada penjelasan rangkaian dimmer lampu flourescent. Untuk keadaan dimana hasil sample lebih besar dari range kontrol maksimum sebesar 3000 lux maka motor stepper yang menarik peneduh akan diaktifkan. Hal ini dimaksudkan untuk sedapat mungkin mengurangi intensitas cahaya yang masuk. Adapun logika pemrogramannya ditunjukkan dalam bentuk diagram alir pada gambar 3.28.



GAMBAR 3.28

ALUR KONTROL BESARAN INTENSITAS CAHAYA

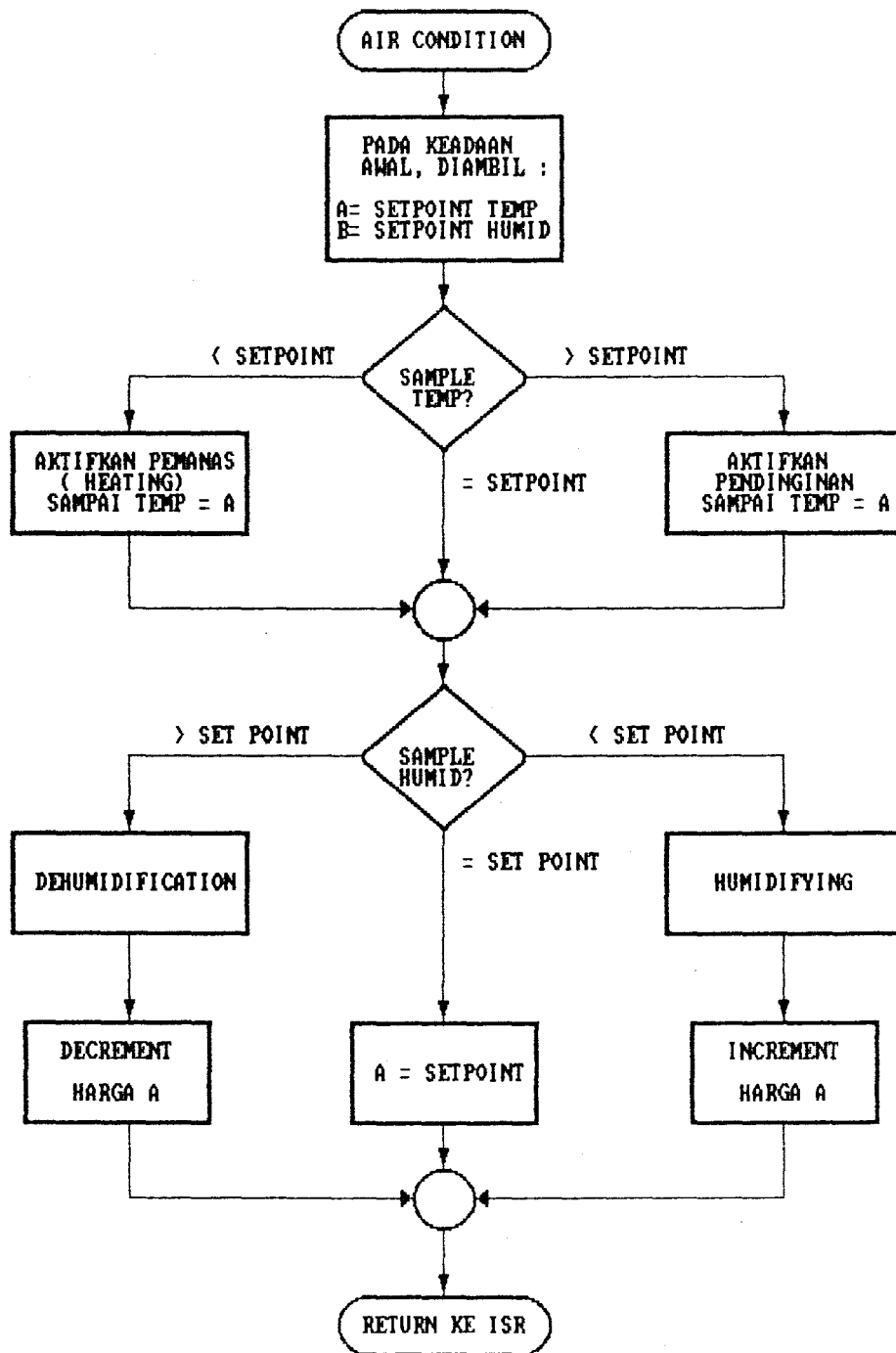
Sedangkan untuk pengaturan besaran temperatur dan kelembaban relatif, uraian mengenai air conditioning pada teori penunjang dapat dijadikan dasar pemikiran. Proses untuk menaikkan dan penurunan temperatur, menaikkan dan menurunkan kelembaban relatif agar sesuai dengan range setting disesuaikan dengan uraian mengenai proses heating, humidifying, cooling dan dehumidification.

Bila temperatur hasil sample sesuai dengan set point maka tidak perlu dilakukan aksi kontrol. Tapi bila temperatur hasil sample masih di bawah set point, maka dilakukan kontrol umpan balik berupa PID kontroler untuk mengaktifkan rangkaian pengatur pemanas. Dengan ini terjadi pemanasan yang akan menaikkan temperatur ruangan. Bila pemanasan telah mencapai set point maka rangkaian pemanas dimatikan. Sebaliknya bila temperatur hasil sample di atas set point, dilakukan proses pendinginan (pendingin yang digunakan pada sistem yang dibuat ini adalah air) agar temperatur turun. Bila temperatur telah mencapai set point maka proses pendinginan dihentikan.

Efek pemanasan di atas umumnya menyebabkan penurunan kelembaban relatif sedangkan pendinginan menyebabkan kenaikan kelembaban relatif udara. Bila hasil sample kelembaban relatif ternyata sesuai dengan set point maka tidak perlu lagi diambil aksi kontrol. Tetapi bila hasil sample kelembaban relatif masih di bawah set point maka

proses humidifying dilakukan yaitu dengan menyemprotkan butiran-butiran air ke ruangan sampai mencapai set point kelembaban relatif yang diinginkan. Bila sebaliknya maka perlu dilakukan dehumidification, yang pada prinsipnya merupakan proses pendinginan. Hal ini mengakibatkan temperatur turun lagi sehingga tidak lagi pada set point. Solusi yang diambil yaitu dengan mengulang kembali proses diatas tetapi dengan pencapaian pemanasan yang bertambah, yaitu di atas set point. Diharapkan saat humidifying atau pendinginan, akan didapat titik temu hasil sample temperatur dan kelembaban relatif dengan set point.

Sebaliknya bila set point di bawah kondisi awal maka dilakukan proses cooling sampai mencapai set point temperatur yang diharapkan. Kemudian apabila hasil sample kelembaban relatif masih di bawah set point, proses cooling terus dilakukan sampai di bawah temperatur dew pointnya. Hal ini juga mengakibatkan temperatur terus turun karenanya proses diulang kembali dengan pencapaian temperatur di atas set point saat cooling untuk mencapai titik temu. Namun apabila hasil ternyata di atas set point, dilakukan proses dehumidification. Selanjutnya proses diulang dengan pencapaian temperatur di bawah set point, kemudian dilakukan pemanasan untuk mencapai titik temu. Algoritma pemrograman uraian di atas ditunjukkan pada gambar 2.29.



GAMBAR 3.29

ALUR KONTROL BESARAN TEMPERATUR DAN KELEMBABAN

BAB IV

PENGUJIAN DAN PENGUKURAN

Terkait dengan pengambilan besaran fisis seperti temperatur, kelembaban dan intensitas cahaya maka sensor yang dipakai harus dikalibrasi agar pengambilan data yang dilakukan benar-benar merupakan besaran yang diukur atau setidaknya masih dalam range yang mendekati.

IV.1. PENGUJIAN SENSOR TEMPERATUR

IC LM335 yang dipakai sebagai sensor temperatur memiliki linearitas yang baik. Kalibrasi terhadap satu titik akan berlaku untuk semua temperatur. Pada pengujian, sensor dikalibrasikan pada temperatur 27°C (suhu kamar). Hasil pengukuran yang dihasilkan dibandingkan dengan suatu termohigrometer sebagai acuan.

Tabel 4-1 memperlihatkan hasil pengujian yang dilakukan. Terlihat bahwa ada kesalahan rata-rata sekitar 0.225 untuk rentang pengukuran 23°C - 32°C. Maka pengukuran temperatur yang dilakukan memiliki akurasi sebesar :

$$\begin{aligned}\text{akurasi} &= 100 \times (0.225/9) \% \\ &= 0.4\%\end{aligned}$$

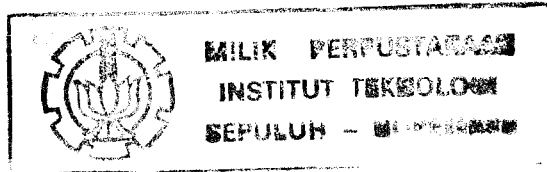
TABEL 4-1.
HASIL PENGUJIAN SENSOR TEMPERATUR

Termo- meter (°C)	Tegangan pada ADC (Volt)				Pembacaan (°C)				Rata- rata (°C)
	T1	T2	T3	T4	T1	T2	T3	T4	
23	0.472	0.454	0.472	0.457	24	23	24	23	23.5
24	0.496	0.470	0.472	0.457	25	24	24	23	24.0
25	0.498	0.492	0.492	0.474	25	25	25	24	24.75
26	0.525	0.512	0.514	0.497	27	26	26	25	26.0
27	0.551	0.518	0.532	0.514	28	26	27	26	26.75
28	0.570	0.552	0.552	0.530	29	28	28	27	28.0
29	0.586	0.572	0.574	0.572	30	29	29	29	29.25
30	0.598	0.578	0.592	0.578	30	29	30	29	29.5
31	0.631	0.612	0.614	0.600	32	31	31	30	31.0
32	0.654	0.630	0.638	0.612	33	32	32	31	32.0

IV.2. PENGUJIAN SENSOR KELEMBABAN

Metoda yang dipakai untuk mengukur kelembaban relatif adalah dengan psychrometer. Metoda ini menggunakan dua buah sensor temperatur dengan pengkondisian yang berbeda Satu daripadanya dibungkus dengan suatu sumbu basah, disebut *wet bulb* dan satunya disebut *dry bulb*.

Kalibrasi dilakukan seperti halnya kalibrasi terhadap sensor temperatur. Dalam pengujian, dilakukan



pembacaan temperatur pada dry bulb dan wet bulb. Untuk konversinya ke besaran kelembaban relatif digunakan dua cara alternatif berdasarkan pembacaan dry bulb dan wet bulb.

Cara yang pertama adalah berdasarkan data tabel yang diturunkan dari psychrometric chart (dapat dilihat pada lampiran). Hasil pembacaan dry bulb dengan hasil pengurangan pembacaan dry bulb dengan wet bulb akan menentukan besarnya kelembaban relatif yang diukur.

Sedangkan cara yang kedua adalah dengan rumus persamaan persentase kelembaban relatif menggunakan tekanan parsial, yaitu : 13)

$$H_R = 100 \times \frac{P_A}{P_{AS}}$$

dimana :

- H_R = persentase kelembaban relatif
- P_A = tekanan parsial uap air dalam percampuran uap air dan udara
- P_{AS} = tekanan uap dari air murni pada temperatur tertentu

Dalam hal ini, harga P_A dan P_{AS} dicari dari steam tabel (pada lampiran) pada temperatur wet bulb dan dry bulb. Karena pada steam tabel tersebut hanya diberikan nilai untuk titik-titik data temperatur tertentu maka untuk

13) Geankoplis, Christie J., *Transport Processes and Unit Operations*, Allyn and Bacon, Inc., USA, hal.514.

mengestimasi nilai antara di antara titik-titik data yang tepat digunakan metoda interpolasi polinom. Interpolasi polinom yang dipakai adalah polinom interpolasi terbagi Newton dengan rumusan umumnya :

$$f_n(T) = f(T_0) + f[T, T_0] (T-T_0) + f[T_2, T_1, T_0] (T-T_0) (T-T_1) + \dots + f[T_n, T_{n-1}, \dots, T_0] (T-T_0) (T-T_1) \dots (T-T_n)$$

Dengan mengambil titik-titik data temperatur yang sesuai dengan range, dalam hal ini dipakai range 15°C - 40°C dari steam tabel, maka berdasarkan perhitungan, polinom interpolasi orde ke-3 sudah cukup akurat untuk estimasi.

Didapat :

$$f[T_0] = 1.7051$$

$$f[T_1, T_0] = 0.119633$$

$$f[T_2, T_1, T_0] = 0.003561$$

$$f[T_3, T_2, T_1, T_0] = 0.000067$$

Jadi persamaan polinom interpolasi orde 3 didapat :

$$f[T] = 1.7051 + 0.119633(T-15) + 0.003561(T-15)(T-18) + 0.000067(T-15)(T-18)(T-21)$$

Persamaan persentase kelembaban relatif menjadi :

$$\therefore H_R = 100 \times \frac{f[T_{\text{wet}}]}{f[T_{\text{dry}}]}$$

Adapun hasil pengukuran untuk besaran kelembaban relatif ditabelkan pada tabel 4-2. Jika dicari regresi linear untuk nilai rata-rata pembacaan kelembaban relatif terhadap pembacaan alat ukur, didapat :

$$y = 7.351 + 0.916 x$$

Deviasi atau penyimpangan hasil pengukuran sensor terhadap nilai standar alat ukur untuk setiap titik pengukuran dapat dihitung dengan rumus :

$$d = | 7.351 + 0.916 x - y |$$

Kesalahan pengukuran sensor temperatur diambil dari penyimpangan terbesar atau deviasi maksimum dari titik-titik pengukuran tersebut. Dari perhitungan, diperoleh deviasi terbesar adalah 3 yaitu pada titik pengukuran 86%. Dengan demikian dalam range pengukuran 55% - 90% maka sensor memiliki akurasi sebesar $3/35 \times 100\% = 8.57\%$.

IV.3. PENGUJIAN SENSOR CAHAYA

Sensor yang dipakai untuk mengukur besarnya intensitas cahaya adalah jenis photovoltaic berupa solar cell. Proses kalibrasi dilakukan dengan acuan dari Luxmeter HIOKI yang mempunyai range pengukuran maksimum 3000 lux.

Pertama dilakukan pengukuran untuk menentukan karakteristik dan linearitas dari sensor yang digunakan. Hasil ini ditabelkan pada tabel 4-3. Terlihat bahwa

sensor masih linear sampai range 3000 lux, dengan arus short circuit sebesar 540 μA .

TABEL 4-2.

HASIL PENGUJIAN SENSOR KELEMBABAN RELATIF

Hygro- meter (%)	Termometer (°C)				Tabel Psychrometric (%)		Rata-rata pembacaan (%)
	DRY1	WET1	DRY2	WET2	HUMID1	HUMID2	
90	24	23	24	23	86	86	86
84	24	22	24	23	82	86	84
76	24	21	24	21	78	78	78
74	25	22	25	21	74	71	73
74	26	22	26	23	68	75	72
72	28	24	28	24	72	72	72
72	30	25	30	26	66	73	70
60	31	25	31	25	61	61	61
58	32	25	32	25	55	55	55
55	33	25	33	25	49	49	49

Kalibrasi dilakukan dengan mencari hubungan antara arus short circuit dengan intensitas cahaya. Hubungan tersebut dicari dengan menurunkan suatu fungsi hampiran yang secara cukup cocok dengan gejala umum data di atas. Pendekatan yang dipakai adalah menggunakan regresi linier

TABEL 4-3
 HUBUNGAN ARUS HUBUNG SINGKAT DENGAN INTENSITAS CAHAYA
 SOLAR CELL

No.	Lx (Lux)	I _{sc} (μ A)
1	110	20
2	195	35
3	300	56
4	400	79
5	550	105
6	620	120
7	750	144
8	825	160
9	950	183
10	1125	215
11	1260	241
12	1350	256
13	1500	267
14	1650	287
15	1770	307
16	1850	319
17	2000	344
18	2140	368
19	2200	382
20	2350	404
21	2520	434
22	2650	456
23	2750	464
24	3000	540

kuadrat terkecil, dimana rumus umum persamaan garisnya adalah :

$$Y = A_0 + A_1 X$$

dimana :

$$A_1 = \frac{n\sum XY - \sum X \sum Y}{n\sum X^2 - (\sum X)^2}$$

$$A_0 = \bar{Y} - A_1 \bar{X}$$

dengan \bar{X} dan \bar{Y} adalah rata-rata (mean) dari X dan Y.

Berdasar data pada tabel 4-3 maka didapatkan hubungan antara besarnya intensitas cahaya dengan arus short circuit yaitu :

$$L_x = -69.2 + 5.9I_{sc}$$

Ini berarti kenaikan arus sebesar 1 μA mengindikasikan kenaikan intensitas cahaya sekitar 5.9 Lux. Dengan menggunakan tahanan 9.5 K Ω pada rangkaian pengubah arus ke tegangan maka kenaikan intensitas cahaya sekitar 5.9 Lux tadi menghasilkan perubahan tegangan sebesar 9.5 mV. Dengan resolusi ADC sebesar 20 mV per step maka kenaikan satu bit menandakan kenaikan intensitas cahaya sebesar $5.9 \times (20/9.5) = 12.42 \text{ Lux} \approx 12 \text{ Lux}$.

Tabel 4-4 menunjukkan hasil pengukuran intensitas cahaya setelah dikalibrasi. Jika dicari regresi linear untuk nilai rata-rata pembacaan besarnya intensitas

cahaya terhadap pembacaan alat ukur, didapat :

$$y = -0.911 + 0.998 x$$

Deviasinya dihitung dari :

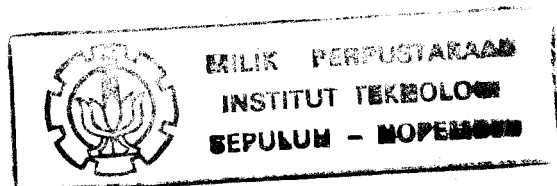
$$d = | -0.911 + 0.998 x - y |$$

Dari perhitungan, diperoleh deviasi terbesar adalah 51 yaitu pada titik pengukuran 1968 lux. Dengan demikian dalam range pengukuran 170 - 2850 lux maka sensor memiliki akurasi sebesar $51/2680 \times 100\% = 1.94\%$.

TABEL 4-4

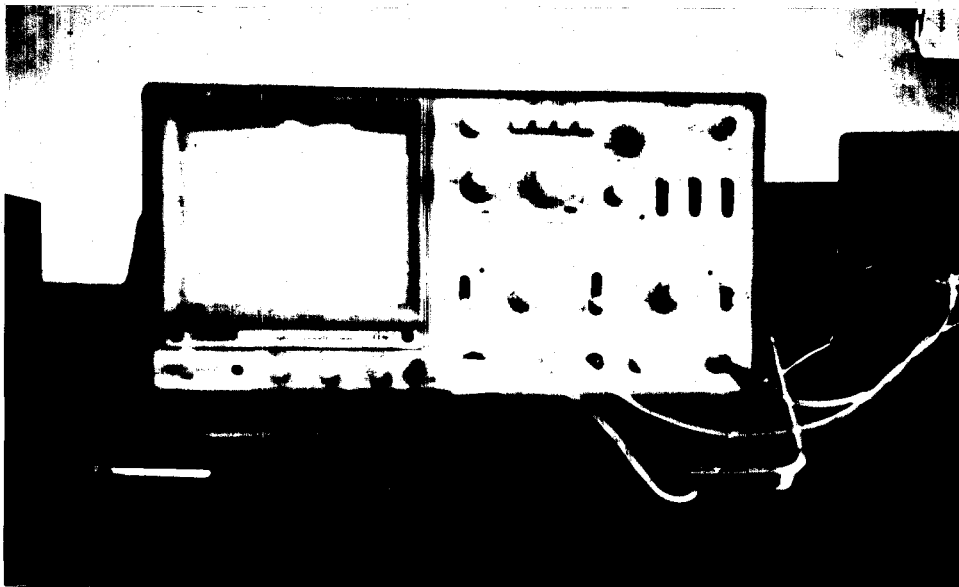
HASIL PENGUJIAN SENSOR CAHAYA

Luxmeter (Lux)	Tegangan masukan ADC (V)	Hasil di monitor (Lux)
170	0.28	168
320	0.56	336
500	0.84	504
600	0.96	588
680	1.08	672
750	1.22	744
1220	2.08	1260
1300	2.19	1322
1650	2.75	1632
1800	2.93	1764
1920	3.31	1968
2350	3.79	2304
2500	4.10	2484
2700	4.53	2724
2850	4.82	2880



IV.4. PENGUJIAN TERHADAP RANGKAIAN TRIGGER

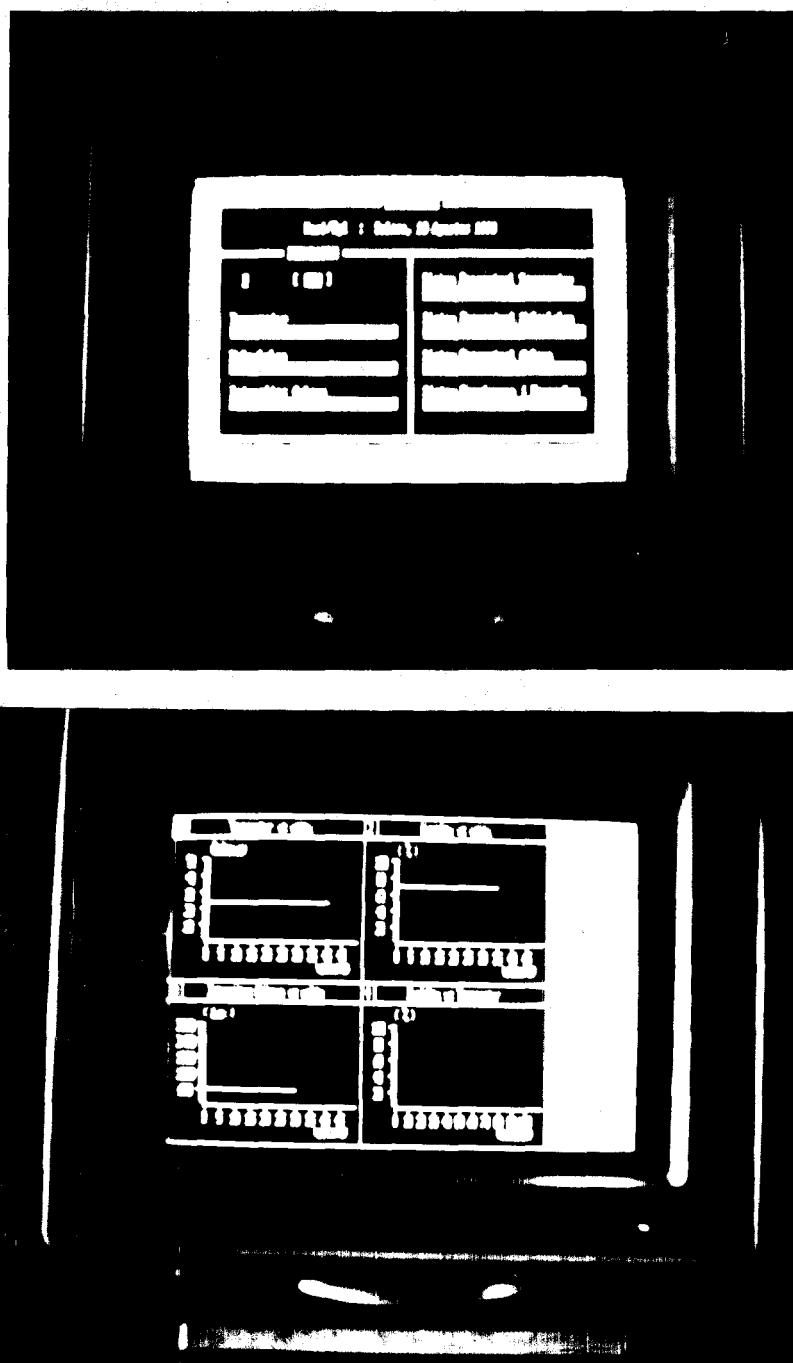
Untuk kontrol kontinyu, PIT 8253 memegang peranan dalam hal phase kontrol. Untuk sinkronisasi dengan tegangan jala-jala maka diperlukan rangkaian trigger yang mana keluarannya dipakai untuk mentrigger gate pada 8253. Pewaktuan trigger akan menentukan sudut penyalaan triac pada rangkaian driver umpan balik. Gambar 4.1 memperlihatkan sinyal penyearahan gelombang penuh dan sinyal trigger yang dihasilkan rangkaian zero crossing detector.



GAMBAR 4.1

PENYEARAHAN TEGANGAN JALA-JALA DAN SINYAL TRIGGER

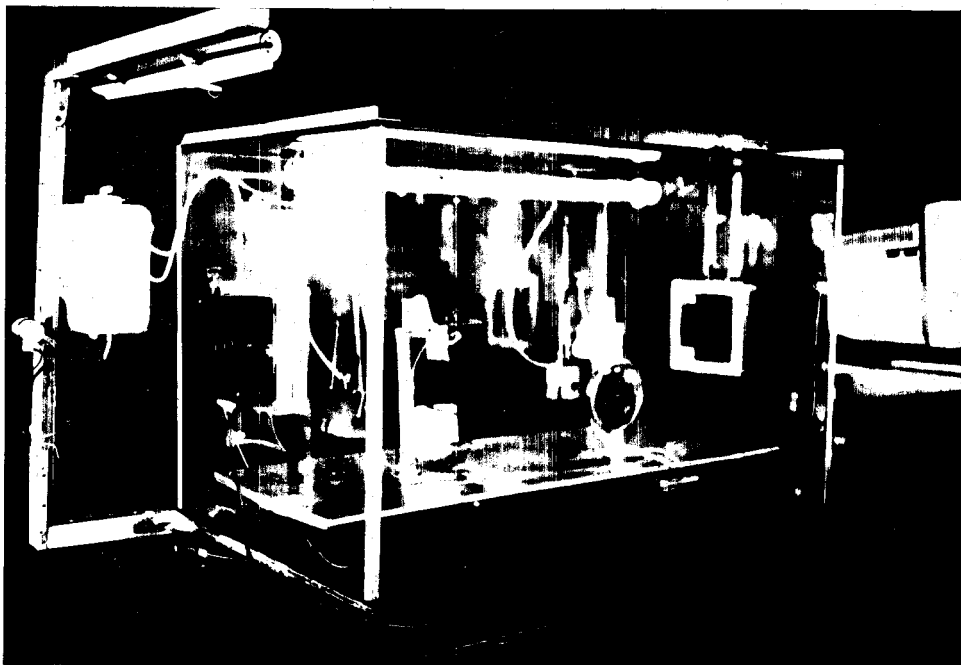
IV.5. FASILITAS PADA PERANGKAT LUNAK



GAMBAR 4.2

TAMPILAN FASILITAS PERANGKAT LUNAK

IV.6. PERANGKAT ELEKTRONIK DAN PROTOTYPE ALAT YANG DIJUAL



GAMBAR 4.3

PERANGKAT KERAS PENALATAN YANG DIJUAL

BAB V

PENUTUP

V.1. KESIMPULAN

Setelah dilakukan pembahasan hingga pengujian dan pengukuran pada sistem kontrol yang dibuat, maka dapat ditarik kesimpulan sebagai berikut :

1. Berdasar hasil pengujian dan pengukuran pada sistem maka didapat bahwa akurasi dari sensor temperatur dalam range pengukuran 23°C - 32°C sebesar 0.4%, akurasi dari sensor kelembaban relatif dalam range pengukuran 55% - 90% sebesar 8.57% dan akurasi dari sensor cahaya dalam range pengukuran 170 lux - 2850 lux sebesar 1.94%. Dalam aplikasinya untuk budi daya anggrek, akurasi pengukuran besaran tersebut masih diijinkan sehingga cukup laik untuk diaplikasikan.
2. Kontroler PID yang diimplementasikan untuk sistem kontrol kontinyu memberikan andil yang besar dalam sistem kontrol yaitu menghasilkan respon yang baik dan cukup cepat terhadap besaran yang diukur.
3. Range pengukuran intensitas cahaya maksimum sebesar 3000 lux dan range kontrol intensitas cahaya (0 - 1350 lux), membatasi lingkup pemakaian peralatan dan juga terhadap jenis anggrek yang dibudidayakan.
4. Tampilan grafik, monitoring terhadap pengukuran

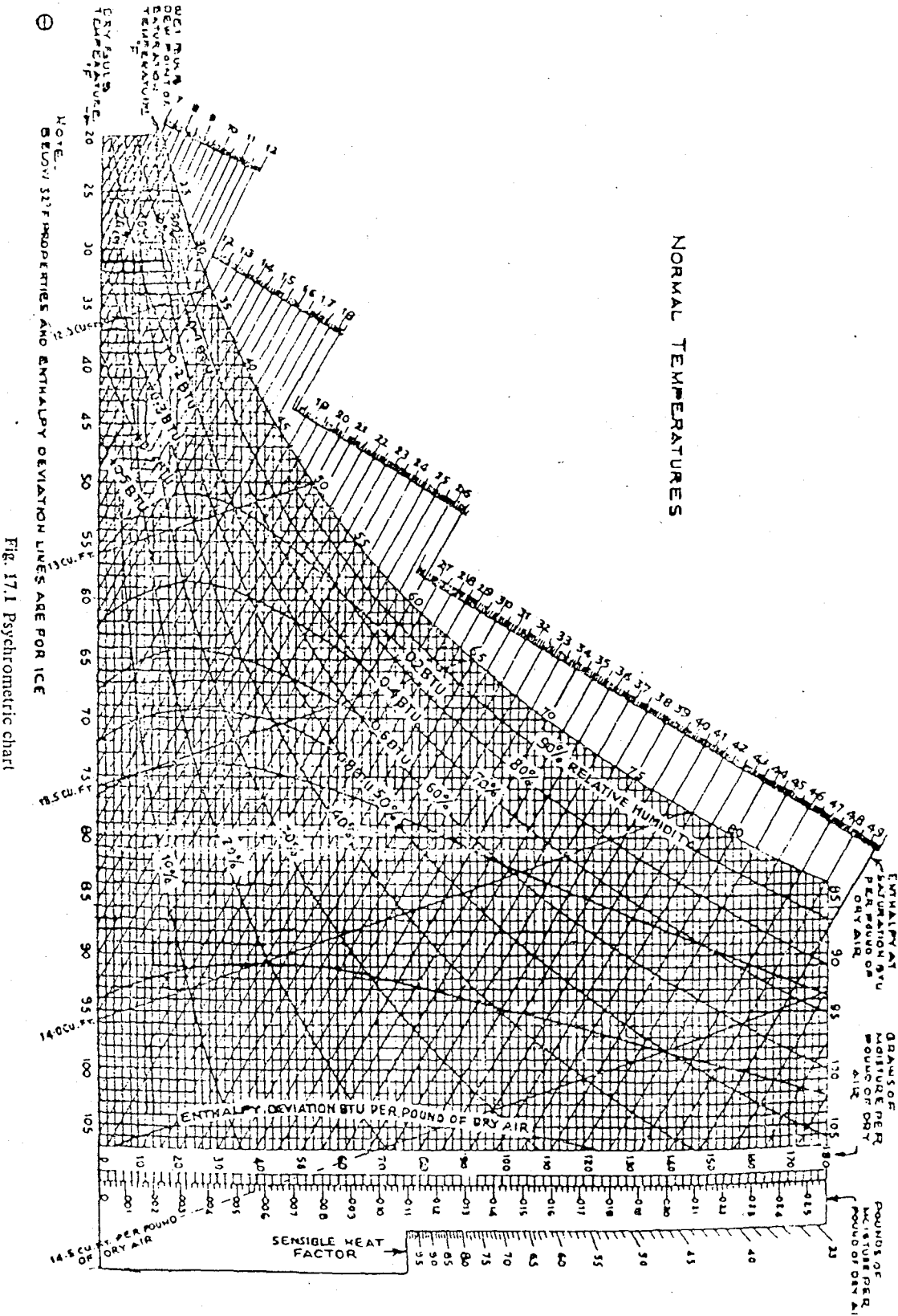
besaran dan kondisi peralatan sebagai kesatuan serta fasilitas pustaka pada perangkat lunak yang dibuat memberikan kemudahan dalam pengoperasian.

V.2. SARAN

1. Untuk mendapatkan unjuk kerja yang lebih baik maka dapat dilakukan penelitian yang lebih mendalam mengenai karakteristik ruangan dan kontrol intensitas cahaya. Dengan demikian akan bisa didapatkan persamaan kontroler yang lebih baik yang bersama-sama dengan PID kontroler memberikan respon umpan balik yang lebih akurat.
2. Pengoperasian peralatan hendaknya tidak dalam ruangan terbuka yang kontak langsung dengan cahaya matahari. Sebabnya peralatan didesain untuk aplikasi dalam ruangan yang memiliki intensitas cahaya yang tidak terlalu kuat. Untuk kebutuhan pengaturan intensitas cahaya yang lebih besar maka harus juga diperhatikan linearitas sensor. Selain itu range kontrol dapat diperluas, misalnya dengan menambah jumlah lampu flourescent.
3. Fleksibelitas dalam pemeliharaan ditekankan pada hal-hal yang dapat dikontrol secara elektronis. Adanya keperluan lain dalam budi daya, seperti pengendalian hama dan penyakit, maka masih perlu penanganan secara konvensional dengan bantuan manusia.

DAFTAR PUSTAKA

- Ananthanarayanan, P.N, *Basic Refregeration and Air Conditioning*, McGraw-Hill, Inc., New Delhi, 1982.
- Coughin, Robert F. , Frederic F. Driscoll, *Operational Amplifiers and Linear Integrated Circuits*, Second Edition, Prentice-Hall, Inc., USA, 1982.
- Eggebrecht, Lewis C., *Interfacing to the IBM Personal Computer*, Howard W. Sams & Co., Inc., First Edition, USA, 1987.
- Fribance, Austin E., *Industrial Instrumentation Fundamentals*, McGraw-Hill, Inc., New Delhi, 1981.
- Gottlieb, Irving M. , *Power Control with Solid State Device*, Reston Publishing Company, Inc., Reston, Virginia, 1985.
- Gunawan, Livy W., *Budidaya Anggrek*, Cetakan ke-4, Penebar Swadaya, Jakarta, 1990.
- Hall, Douglas V., *Microprocessors and Interfacing*, McGraw Hill, Inc., Singapore, 1986.
- Humpries, James T., Leslie P. Sheets, *Industrial Electronics*, Breton Publishers, Massachusetts, USA, 1983.
- Looho, Aloysius, *Penuntun Cara Memelihara Anggrek*, Anggrek Surabaya, Surabaya, 1978.
- Maloney, Timothy J. , *Industrial Solid-State Electronics*, Second Edition, Prentice-Hall, Inc., New Jersey, USA, 1986.
- McAdams, William H., *Heat Transmission*, Third Edition, McGraw-Hill, Inc., Singapore, 1985.
- Schuler, Charles A. , William L. McNamee , *Industrial Electronics and Robotics*, First Edition, McGraw-Hill, Inc., Singapore, 1986.
- , *Optoelectronics Device Data*, Motorola, Inc., USA, 1988.



TABEL PEMBACAAN KELEMBABAN RELATIF

LAMPIRAN A-2

(Difference between readings of wet and dry bulbs, °F)

Dry bulb minus wet bulb, °F:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Dry bulb, °F	Per cent of saturation																													
40	92	83	73	63	50	33	18	7	3	0																				
41	92	84	74	64	51	34	19	8	4	1																				
42	92	85	75	65	52	35	20	9	5	2																				
43	92	85	77	70	63	55	48	42	35	28	21	14	8	1																
44	93	85	78	71	63	56	49	43	36	30	23	16	10	4																
45	93	86	78	71	64	57	51	44	38	31	25	18	12	6	3															
46	93	86	79	72	65	58	52	45	39	32	26	20	14	8	2															
47	93	86	79	72	66	59	53	46	40	34	28	22	16	10	6	1														
48	93	86	79	73	66	60	54	47	41	35	29	23	18	12	7	3														
49	93	86	80	73	67	61	54	48	42	36	31	25	19	14	9	3														
50	93	87	80	74	67	61	55	49	43	38	32	27	21	16	10	6	2													
51	94	87	81	75	68	62	56	50	45	39	34	28	23	17	12	7	3	0												
52	94	87	81	75	69	63	57	51	46	40	35	29	24	19	14	9	4	1												
53	94	87	81	75	69	63	58	52	47	41	36	31	26	20	16	10	6	3	1											
54	94	88	82	76	70	64	59	53	48	42	37	32	27	22	17	12	8	3	3											
55	94	88	82	76	70	65	59	54	49	43	38	33	28	23	19	14	9	5	0											
56	94	88	82	77	71	65	60	55	50	44	39	34	30	25	20	16	11	7	2	0										
57	94	88	82	77	71	66	61	56	51	45	40	35	31	26	22	17	13	8	4	0										
58	94	88	83	77	72	66	61	56	51	46	41	37	32	27	23	18	14	10	6	3										
59	94	89	83	78	73	67	62	57	52	47	42	38	33	29	24	20	16	11	7	3										
60	94	89	83	78	73	68	63	58	53	48	43	39	34	30	26	21	17	13	9	5	1									
61	94	89	84	78	73	68	63	58	54	49	44	40	35	31	27	23	18	14	10	7	3									
62	94	89	84	79	74	69	64	59	54	50	45	41	36	32	28	24	20	16	12	8	4	1								
63	95	89	81	79	74	69	64	60	55	50	46	42	37	33	29	25	21	17	13	10	6	4	2							
64	95	90	84	79	74	70	65	60	56	51	47	43	38	34	30	26	22	18	15	11	7	4	0							
65	95	90	85	80	75	70	66	61	56	52	48	44	40	36	32	29	25	21	17	14	9	5	2							
66	95	90	85	80	75	71	66	62	58	53	49	45	41	37	33	30	26	22	19	16	12	8	5	0						
67	95	90	85	80	76	71	67	62	58	54	50	46	42	38	34	31	27	23	20	18	13	9	6	3						
68	95	90	85	81	76	72	67	63	59	55	51	47	43	39	35	32	28	24	21	18	14	11	8	5	1					
69	95	90	85	81	76	72	67	63	59	55	51	47	43	39	35	32	28	24	21	18	14	11	8	5	1					
70	95	90	86	81	77	72	68	64	60	56	52	48	44	40	36	33	29	25	22	19	16	13	9	6	3					
71	95	90	86	81	77	72	68	64	60	56	52	48	45	41	37	33	30	27	23	20	17	13	10	7	4					
72	95	91	86	82	77	73	69	65	61	57	53	49	45	42	38	34	31	28	24	21	18	15	12	9	6	3				
73	95	91	86	82	78	73	69	65	61	57	53	50	46	42	39	35	32	29	25	22	19	16	13	10	7	4				
74	95	91	86	82	78	74	69	65	61	58	54	50	47	43	39	36	33	29	26	23	20	17	14	11	8	5	1			
75	96	91	86	82	78	74	70	66	62	58	54	51	47	44	40	37	34	30	27	24	21	18	15	12	9	6	3			
76	96	91	87	82	78	74	70	66	62	59	55	51	48	44	41	38	34	31	28	25	22	19	16	13	11	8	5	1		
77	96	91	87	83	79	74	71	67	63	59	56	52	48	45	42	39	35	32	29	26	23	20	17	14	12	9	6	3		
78	96	91	87	83	79	76	71	67	63	60	56	53	49	46	43	39	36	33	30	27	24	21	18	16	13	10	8	5		
79	96	91	87	83	79	76	71	68	64	60	57	53	50	46	43	40	37	34	31	28	25	22	19	17	14	11	9	6		
80	96	91	87	83	79	76	72	68	64	61	57	54	50	47	44	41	38	35	32	29	26	23	20	18	15	12	10	7		
81	96	92	88	84	80	76	72	69	65	61	58	55	51	48	45	42	39	36	33	30	28	25	22	20	18	15	12	10	7	
82	96	92	88	84	80	76	73	69	66	62	59	56	52	49	46	43	40	37	34	32	29	26	24	21	19	16	14	12	9	
83	96	92	88	84	81	77	73	70	66	63	60	57	53	50	47	44	42	39	36	33	31	28	26	23	21	19	17	15	13	
84	96	92	88	86	81	77	74	70	67	64	61	57	54	51	48	46	43	40	37	35	32	30	27	25	23	21	19	17	15	
85	96	92	89	85	81	78	74	71	68	65	62	58	55	52	49	47	44	41	39	36	34	31	29	26	24	22	20	18	16	
86	96	92	89	85	82	78	75	72	69	66	63	60	57	54	51	48	46	43	40	38	35	33	31	29	27	25	23	21	19	
87	96	93	89	86	82	79	76	73	70	67	64	61	58	56	53	50	47	45	42	40	38	36	34	32	30	28	26	24	22	
88	96	93	89	86	83	79	76	73	70	67	64	61	58	56	53	50	48	45	43	40	38	36	34	32	30	28	26	24	22	
89	96	93	89	86	83	79	76	73	70	67	64	61	58	56	53	50	48	45	43	40	38	36	34	32	30	28	26	24	22	
90	96	93	89	86	83	80	77	74	71	68	65	62	59	56	53	50	48	45	43	40	38	36	34	32	30	28	26	24	22	
91	96	93	90	87	83	80	77	74	71	68	65	62	60	57	55	52	49	47	45	43	41	39	37	35	33	31	29	27	25	
92	96	93	90	87	84	81	78	75	72	69	66	64	61	58	56	53	51	49	46	44	42	40	38	36	34	32	30	28	26	
93	96	93																												

Properties of Saturated Steam and Water (Steam Table),
SI Units

Temperature (°C)	Vapor Pressure (kPa)	Specific Volume (m ³ /kg)		Enthalpy (kJ/kg)		Entropy (kJ/kg·K)	
		Liquid	Sat'd Vapor	Liquid	Sat'd Vapor	Liquid	Sat'd Vapor
0.01	0.6113	0.0010002	206.136	0.00	2501.4	0.0000	9.1562
3	0.7577	0.0010001	138.132	12.57	2506.9	0.0457	9.0773
6	0.9349	0.0010001	137.734	25.20	2512.4	0.0912	9.0003
9	1.1477	0.0010003	113.386	37.80	2517.9	0.1362	8.9253
12	1.4022	0.0010005	93.784	50.41	2523.4	0.1806	8.8524
15	1.7051	0.0010009	77.926	62.99	2528.9	0.2245	8.7814
18	2.0640	0.0010014	55.038	75.58	2534.4	0.2679	8.7123
21	2.487	0.0010020	54.514	88.14	2539.9	0.3109	8.6450
24	2.985	0.0010027	45.883	100.70	2545.4	0.3534	8.5794
25	3.169	0.0010029	43.360	104.89	2547.2	0.3674	8.5580
27	3.567	0.0010035	38.774	113.25	2550.8	0.3954	8.5156
30	4.246	0.0010043	32.894	125.79	2556.3	0.4369	8.4533
33	5.034	0.0010053	28.011	138.33	2561.7	0.4781	8.3927
36	5.947	0.0010063	23.940	150.86	2567.1	0.5188	8.3336
40	7.384	0.0010078	19.523	167.57	2574.3	0.5725	8.2570
45	9.593	0.0010099	15.258	188.45	2583.2	0.6387	8.1648
50	12.349	0.0010121	12.032	209.33	2592.1	0.7038	8.0763
55	15.758	0.0010146	9.568	230.23	2600.9	0.7679	7.9913
60	19.940	0.0010172	7.671	251.13	2609.6	0.8312	7.9096
65	25.03	0.0010199	6.197	272.06	2618.3	0.8935	7.8310
70	31.19	0.0010228	5.042	292.98	2626.8	0.9549	7.7553
75	38.58	0.0010259	4.131	313.93	2635.3	1.0155	7.6824
80	47.39	0.0010291	3.407	334.91	2643.7	1.0753	7.6122
85	57.83	0.0010325	2.828	355.90	2651.9	1.1343	7.5445
90	70.14	0.0010360	2.361	376.92	2660.1	1.1925	7.4791
95	84.55	0.0010397	1.9819	397.96	2668.1	1.2500	7.4159
100	101.35	0.0010435	1.6729	419.04	2676.1	1.3069	7.3549

Source: Abridged from J. H. Keenan, F. G. Keyes, P. G. Hill, and J. G. Moore, *Steam Tables—Metric Units*, New York: John Wiley & Sons, Inc., 1969. With permission of the authors and publishers.

LISTING PROGRAM

LAMPIRAN B-1

```

/* * TITLE : UTIL.H */

#include <conio.h>
#include <stdio.h>

#define tputc(x,y,ch)

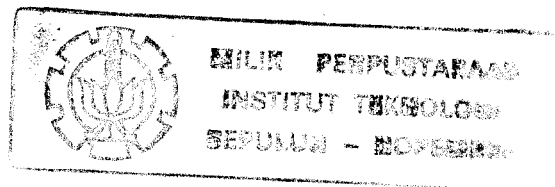
#define wkt " : : "
#define format " - - "

struct orc1 {
    unsigned start : 1;
    unsigned end : 1;
    unsigned setfan : 1;
    unsigned setsiram : 1;
    unsigned setpupuk : 1;
    unsigned temp : 1;
    unsigned humid : 1;
    unsigned fan : 1;
    unsigned light : 1;
    unsigned siram : 1;
    unsigned pupuk : 1;
    unsigned param : 1;
    unsigned balance : 1;
    unsigned mode : 1;
    unsigned : 2;
};

struct orc2 {
    int lama;
    char tm1[9];
    char tm2[9];
    char tm3[9];
};

struct orc3 {
    int intv;
    char tm[9];
    char tgl1[9];
    char tgl2[9];
    char tgl3[9];
    char tgl4[9];
};

```



LISTING PROGRAM

LAMPIRAN B-2

```
struct orc4 {
    unsigned temp;
    unsigned humid;
    unsigned light;
} ;

struct orc5 {
    char time[9];
    unsigned temp1;
    unsigned temp2;
    unsigned temp3;
    unsigned temp4;
    unsigned dry1;
    unsigned wet1;
    unsigned dry2;
    unsigned wet2;
    unsigned sun;
    unsigned humid1;
    unsigned humid2;
    unsigned temp;
    unsigned humid;
    unsigned light;
    unsigned delta;
};

typedef struct orc1 o_flag;
typedef struct orc2 wsiram;
typedef struct orc3 wpupuk;
typedef struct orc4 param;
typedef struct orc5 DataParam;

struct thl {
    DataParam dat;
    struct thl *sbl;
    struct thl *ssd;
};

typedef struct thl addr;
```

```
/* TITLE : KEY.H */
```

```
#include <conio.h>
```

```
#define nocursor()
```

```
#define cursor()
```

```
#define SCREEN
```

```
_setcursortype(_NOCURSOR)
```

```
_setcursortype(_NORMALCURSOR)
```

```
0xb8000000
```

```
#define RUN_MODE 1
```

```
#define STOP_MODE 0
```

```
#define TRUE 1
```

```
#define FALSE 0
```

```
#define ATAS 1
```

```
#define BAWAH 0
```

```
#define UP 72
```

```
#define DOWN 80
```

```
#define RIGHT 77
```

```
#define LEFT 75
```

```
#define ENTER 28
```

```
#define ALTO 24
```

```
#define ALTC 46
```

```
#define ALTN 49
```

```
#define ALTR 19
```

```
#define ALTS 31
```

```
#define ALTI 23
```

```
#define ALTF 33
```

```
#define ALTX 45
```

```
#define ALTQ 16
```

```
#define PgUp 73
```

```
#define PgDn 81
```

```
#define HOME 71
```

```
#define END 79
```

```
#define ESC 1
```

```
#define BKSPC 8
```

```
#define F1 59
```

```
#define F2 60
```

```
#define F3 61
```

```
#define F4 62
```

```
#define F5 63
```

```
#define F6 64
```

```
#define F7 65
```

```
#define F8 66
```

```
#define F10 68
```

```
#define ALT1 120
```

```
#define ALT2 121
```

```
#define ALT3 122
```

```
#define ALT4 123
```

```
/* TITLE : DEVICE.H --> Addressing for device controller */
```

```
/* ----- PPI 8255 #1 ----- */
```

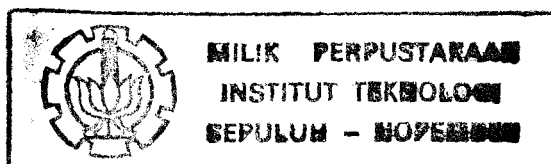
```
#define PA1    0x350  
#define PB1    0x351  
#define PC1    0x352  
#define CW1    0x353
```

```
/* ----- PPI 8255 #2 ----- */
```

```
#define PA2    0x354  
#define PB2    0x355  
#define PC2    0x356  
#define CW2    0x357
```

```
/* ----- PIT 8253 ----- */
```

```
#define CTR0    0x330  
#define CTR1    0x331  
#define CTR2    0x332  
#define CWCTR   0x333
```



```

#include <stdlib.h>
#include <math.h>
#include <string.h>
#include "device.h"
#include <fcntl.h>
#include <alloc.h>
#include <time.h>
#include <io.h>
#include <bios.h>
#include <conio.h>
#include <dos.h>
#include <ctype.h>
#include <stdio.h>
#include <errno.h>
#include <process.h>
#include <graphics.h>
#include <alloc.h>
#include <dir.h>
#include "util.h"
#include "key.h"
#include "glib.h"

/* function prototype */

void interrupt (*oldhandler)();
void Init_var(void);
void InitDevice(void);
unsigned GetSample(unsigned char channel);
void ControlDevice(void);
int GetParam(void);
int erase_awal(addr *node);
int add_data(addr *duri);
addr *search(addr *node, int num);
int conditioning(void);
int ServeLight(void);
void ActivateStepper(unsigned rotate, int direction);
void ResetSystem(void);
int TPID(void);
int LPID(void);
int tmsample(void);
int ConMode(void);
int parameter(void);
int inputnum(int x,int y,int nyala,int attr,int *point);
int getinput(int x,int y,int mode,char *form,int len);
int tgl_pupuk(void);
int waktu_siram(void);
int device(void);

```



```

void All(void);
int  print_data(int x,int y,addr *node, int mode,int attr);
void win_select(int num, int mode);
int  shift_up(int num);
int  shift_down(int num);
int  Detail(void);
void about(void);
int  menu(void);
extern int GraphView(void);
extern int FileInfo(void);

```

```

/* ----- */

```

```

char buf1[4000], buf2[4000];
char dev,rtm[9],rdt[9];
int  pilih1,pilih2,pilih3,delta,t_control;
int  pp,zz,z,u,tsp,count,step,rr,ss,pos,maks,ci;
int  t_err,t_preverr,l_err,l_preverr, TPWM0, LPWM0;
float t_int, l_int;
param set,ps,pset;
wpupuk pupuk,setp;
wsiram siram,in;
o_flag f;
addr *awal,*akhir,*dtop,*br_atas;
addr *batas,*bts1,*bts2,*bts3,*bts4;
extern glink *first,*last;
unsigned char lpwm,tpwm;
char tstr[10],hstr[10],lstr[10];

```

```

void Init_var(void)
{
    awal = akhir = batas = dtop = br_atas = NULL;
    bts1 = bts2 = bts3 = bts4 = NULL;
    first = last = NULL;
    pilih1 = pilih2 = pilih3 = dev = pp = rr = ss = z = 0;
    t_err = t_preverr = l_err = l_preverr = 0;
    t_int = l_int = 0.0;
    tsp = 30; step = 0x3333; count = tsp * CLK_TCK;
    siram.lama = 0;
    strcpy(siram.tm1,wkt);
    strcpy(siram.tm2,wkt);
    strcpy(siram.tm3,wkt);
    in = siram;
    pupuk.intv = 0;
    strcpy(pupuk.tm,wkt);
    strcpy(pupuk.tgl1,format);
    strcpy(pupuk.tgl2,format);
}

```

```

strcpy(pupuk.tgl3,format);
strcpy(pupuk.tgl4,format);
setp = pupuk;
GetParam();
f.start = f.setsiram = f.setpupuk = FALSE;
f.temp = f.humid = f.light = f.siram = f.pupuk = FALSE;
f.end = f.param = f.balance = f.setfan = f.fan = FALSE;
f.mode = 0;
set.temp = awal->dat.temp;
set.humid = awal->dat.humid;
set.light = awal->dat.light;
t_control = set.temp; ps = pset = set;
erase_awal(awal);
dev = 1; outportb(PB2,dev);
}

void InitDevice(void)
{
    outportb(CW1,0x83);
    outportb(CW2,0x80);
    outportb(CWCTR,0x12);
    outportb(CTR0,0xff);
    outportb(CWCTR,0x52);
    outportb(CTR1,0xff);
}

unsigned GetSample(unsigned char channel)
{
    unsigned char eoc,data;
    register int i;

    outportb(PA1,channel | 0x40);
    outportb(PA1,channel | 0xc0);
    outportb(PA1,channel);
    while(eoc) eoc = inportb(PC1) & 1;
    while(!eoc) eoc = inportb(PC1) & 1;
    data = inportb(PB1);
    return((unsigned) data);
}

void interrupt ISR()
{
    strcpy(rtm, (char *) r_time());
    strcpy(rdt, (char *) r_date());
    if(f.start == TRUE)
    {
        if(rr>0 && --rr==0)
            { f.pupuk = FALSE; dev &= 0x0b; outportb(PB2,dev); }
    }
}

```

```

if(ss>0 && --ss==0)
{ f.siram = FALSE; dev &= 0x0b; outportb(PB2,dev); }
if(f.setpupuk == TRUE && f.pupuk == FALSE)
{
if(!strcmp(rdt,pupuk.tgl1) || !strcmp(rdt,pupuk.tgl2) ||
!strcmp(rdt,pupuk.tgl3) || !strcmp(rdt,pupuk.tgl4))
{
if(!strcmp(rtm,pupuk.tm))
{
rr = (unsigned) pupuk.intv * CLK_TCK;
f.pupuk = TRUE;
dev |= 4; outportb(PB2,dev);
}
}
}
if(f.setsiram==TRUE && f.siram == FALSE)
{
if(!strcmp(rtm,siram.tm1) || !strcmp(rtm,siram.tm2) ||
!strcmp(rtm,siram.tm3))
{
ss = siram.lama * CLK_TCK;
f.siram = TRUE;
dev |= 4; outportb(PB2, dev);
}
}
}

if(f.step == TRUE)
{ if(zz++ <= 4200) ActivateStepper(1,0); }

if(count-- < 1)
{
disable();
GetParam();
++z;
if(++ci > 100) { --ci; erase_awal(awal); }
conditioning();
ServeLight();
count = tsp * CLK_TCK;
delta += tsp;
enable();
}
}
oldhandler();
}

```

```

int erase_awal(addr *node)
{
    if(node)
    {
        awal = node->ssd;
        if(awal) awal->sbl = NULL;
        else akhir = NULL;
    }
    return;
}

int add_data(addr *duri)
{
    if(awal == NULL)
    {
        duri->ssd = NULL;
        duri->sbl = NULL;
        awal = duri;
        akhir = duri;
        return;
    }
    else
    {
        akhir->ssd = duri;
        duri->sbl = akhir;
        duri->ssd = NULL;
        akhir = duri;
    }
    return;
}

int GetParam(void)
{
    int  hd;
    addr *dt;

    outportb(PC1,0x80);
    if((dt = (addr *) malloc(sizeof(addr))) == NULL)
    { t_error(err[3]); return; }
    strcpy(dt->dat.time,rtm);
    dt->dat.delta = delta;
    dt->dat.temp1 = GetSample(0);
    dt->dat.temp2 = GetSample(0x09);
    dt->dat.temp3 = GetSample(0x12);
    dt->dat.temp4 = GetSample(0x1b);
    dt->dat.sun   = GetSample(0x06);
    dt->dat.dry1  = GetSample(0x24);
    dt->dat.wet1  = GetSample(0x2c);

```

```

int erase_awal(addr *node)
{
    if(node)
    {
        awal = node->ssd;
        if(awal) awal->sbl = NULL;
        else akhir = NULL;
    }
    return;
}

int add_data(addr *duri)
{
    if(awal == NULL)
    {
        duri->ssd = NULL;
        duri->sbl = NULL;
        awal = duri;
        akhir = duri;
        return;
    }
    else
    {
        akhir->ssd = duri;
        duri->sbl = akhir;
        duri->ssd = NULL;
        akhir = duri;
    }
    return;
}

int GetParam(void)
{
    int hd;
    addr *dt;

    outportb(PC1,0x80);
    if((dt = (addr *) malloc(sizeof(addr))) == NULL)
    { t_error(err[3]); return; }
    strcpy(dt->dat.time,rtm);
    dt->dat.delta = delta;
    dt->dat.temp1 = GetSample(0);
    dt->dat.temp2 = GetSample(0x09);
    dt->dat.temp3 = GetSample(0x12);
    dt->dat.temp4 = GetSample(0x1b);
    dt->dat.sun = GetSample(0x06);
    dt->dat.dry1 = GetSample(0x24);
    dt->dat.wet1 = GetSample(0x2c);
}

```

```

dt->dat.dry2    = GetSample(0x35);
dt->dat.wet2    = GetSample(0x3d);
if(f.mode)
{
    dt->dat.humid1 = bulat(100*equation(dt->dat.wet1)/
                                equation(dt->dat.dry1));
    dt->dat.humid2 = bulat(100*equation(dt->dat.wet2)/
                                equation(dt->dat.dry2));
}
else
{
    dt->dat.humid1 = bacatabel(dt->dat.dry1,dt->dat.wet1);
    dt->dat.humid2 = bacatabel(dt->dat.dry2,dt->dat.wet2);
}
dt->dat.temp    = bulat((dt->dat.temp1+dt->dat.temp2+dt->dat.temp3+
                        dt->dat.temp4)/4.);
dt->dat.humid   = bulat((dt->dat.humid1 + dt->dat.humid2)/2.);
dt->dat.light   = dt->dat.sun * 12;
add_data(dt);
f.param = TRUE;
outportb(PC1,0);
return;
}

int conditioning(void)
{
    if(f.balance == FALSE)
    {
        if(t_control > akhir->dat.temp)
        { f.temp = TRUE; tpwm = TPID(); outportb(CTRO,tpwm); }
        else if(t_control < akhir->dat.temp)
        {
            if(f.humid == FALSE)
            { dev != 0x0a; outportb(PB2,dev); f.humid = TRUE; }
        }
        else
        {
            f.balance = TRUE;
            if(f.temp == TRUE) { f.temp = FALSE; outportb(CTRO,0xff); }
            if(f.humid == TRUE)
            { f.humid = FALSE; dev &= 5; outportb(PB2,dev); }
        }
    }
    else
    {
        if(akhir->dat.temp != set.temp)
        {
            if(f.dehumid == TRUE) t_control--;
        }
    }
}

```

```

        else if(f.humid == TRUE) t_control++;
        else t_control = set.temp;
        dev &= 5; outportb(PB2,dev);
        f.balance = FALSE;
        return;
    }
    else
    {
        if(akhir->dat.humid < set.humid-6)
        { f.humid = TRUE; dev != 0x0a; outportb(PB2,dev); }
        else if(akhir->dat.humid > set.humid+6)
        { f.dehumid = TRUE; dev != 0x0a; outportb(PB2,dev); }
    }
}
return;
}

int ServeLight(void)
{
    if(akhir->dat.light > set.light-37 && akhir->dat.light <
        set.light+37)
        f.light = FALSE;
    else
        if(awal && akhir->dat.light > 3000) f.step = TRUE;
    else
    {
        f.light = TRUE;
        lpwm = LPID(); outportb(CTR1,lpwm); outportb(PC1,0);
    }
    return;
}

void ActivateStepper(unsigned rotate, int direction)
{
    register int i;

    for(i=0;i<rotate;i++)
    {
        outportb(PA2,(unsigned char)step); delay(15);
        direction == 0 ? (step = _rotr(step,1)) :
        (step = _rotr(step,1));
    }
}

```

```

int TPID(void)
{
    float H;
    int VarT;

    t_preverr = t_err;
    t_err = t_control - akhir->dat.temp;
    t_int += 0.001 * t_err * tsp;
    H = 1.0 * t_err + t_int + 3.0 * (t_err - t_preverr)/tsp;
    VarT = bulat(80 - H * 4);
    return(VarT);
}

int LPID(void)
{
    float H;
    int VarL;

    if(l_err == 0)
    {
        if(set.light > 1350) LPWM0 = 85;
        else
            LPWM0 = 170.3 - 0.071 * set.light;
    }
    l_preverr = l_err;
    l_err = set.light/12 - akhir->dat.sun;
    l_int += 0.005 * l_err * tsp;
    H = 0.5 * l_err + l_int + 0.5 * (l_err - l_preverr)/tsp;
    VarL = LPWM0 - bulat(15.036 + 0.854 * H);
    return(VarL);
}

void ResetSystem(void)
{
    char *buffer;

    if(f.start == TRUE) f.start = FALSE;
    outportb(PB2,0);
    outportb(CTR0,0xff);
    outportb(CTR1,0xff);
    if(zz)
    {
        if((buffer = (char *) malloc(2000)) == NULL) return;
        gettext(20,10,60,15,buffer); nocursor();
        textattr(0x4f); dbox(20,10,59,14,double_line);
        shadow(20,10,59,14);
        gotoxy(38,11); cprintf(" ERROR ! ");
    }
}

```



```

        cursor(); textattr(0x20);
        textcolor(14+128); gotoxy(35,12); cprintf(" WAIT ... ");
        while(zz--) ActivateStepper(zz,1);
        puttext(20,10,60,15,buffer);
        free(buffer);
    }
    Init_var();
}

int tmsample(void)
{
    sbx(10,7,69,15,double_line,1);
    Status_OK(37,13,k_out[0],bottom,0x30,1);
    textattr(0x1e); gotoxy(15,7); cprintf(" SETTING SAMPLING TIME ");
    cursor(); textattr(0x70); gotoxy(18,10);
    cprintf(" Waktu sampling      : [      ] second      ");
    inputnum(43,10,0x4f,0x70,&tsp);
    count = tsp * CLK_TCK;
    return 0;
}

int ConMode(void)
{
    char ch;
    int mode;

    mode = f.mode; sbx(10,7,69,15,double_line,1);
    Status_OK(37,13,k_out[0],bottom1,0x30,1);
    textattr(0x1e); gotoxy(15,7);
    cprintf(" METODE KONVERSI RELATIF HUMIDITY");
    cursor(); textattr(0x70); gotoxy(18,10);
    cprintf(" Metode Konversi : [      ]      ");
    for(;;)
    {
        ch = select(41,10,cond2,2,&mode,nothing);
        if(ch == ALTO) { f.mode = mode; nocursor(); return; }
        if(ch == ESC) { nocursor(); return; }
    }
}

int inputnum(int x,int y,int nyala,int attr,int *point)
{
    int ts;
    char ch1,ch2,tp[7],KEY[5];

    sprintf(KEY,"%c%c%c%c%c",UP,DOWN,ESC,ENTER,ALTO);
    ts = *point;

```

```

for(;;)
{
    sprintf(tp," %-4d ",*point);
    textattr(nyala); nocursor(); gotoxy(x,y); cprintf("%s",tp);
    do { baca(&ch1,&ch2); } while(!strchr(KEY,ch2));
    if(ch2 == ESC ) { *point = ts; return(ch2); }
    if(ch2 == DOWN || ch2 == UP)
    {
        textattr(attr); cursor(); gotoxy(x,y); cprintf("%s",tp);
        return(ch2);
    }
    if(ch2 == ALTO) return (ch2);
    if(ch2 == ENTER)
    {
        textattr(attr); cursor(); space(x,y,strlen(tp));
        ch2 = inputint(x+1,y,tp,4);
        if(ch2 == ESC)
        {
            textattr(nyala); nocursor(); gotoxy(x,y); cprintf("%s",tp);
            return 0;
        }
        *point = atoi(tp);
    }
}
}

int parameter(void)
{
    int tc = 1;
    char ch;

    sprintf(tstr," [%-2d] ",set.temp);
    sprintf(hstr," [%-2d] ",set.humid);
    sprintf(lstr," [%-4d] ",set.light);
    ps = set;
    sbbox(13,6,68,19,double_line,1);
    Status_OK(37,17,k_out[0],bottom,0x30,1);
    textattr(0x1e); gotoxy(18,6); cprintf(" SETTING SET POINT ");
    textattr(0x1f); gotoxy(13,8);
    cprintf("||===== BESARAN ===== SET POINT =====||");
    cursor(); textattr(0x70); gotoxy(19,10);
    cprintf(" Temperatur          :      [%s] °C      ",tstr);
    gotoxy(19,12);
    cprintf(" Kelembaban relatip    :      [%s] %      ",hstr);
    gotoxy(19,14);
    cprintf(" Intensitas cahaya     :      [%s] Lux      ",lstr);

```



```

        break;

        case 2 : ps.light = atoi(a);
                sprintf(form," [%-4d] ",ps.light);
                break;
    }
    textattr(0x2f); nocursor(); gotoxy(x,y); cprintf("%s",form);
    do { baca(&ch1,&ch2); } while(!strchr(KEY,ch2));
    textattr(0x70); gotoxy(x,y); cprintf("%s",form);
    return(ch2);
}
}
}
textattr(0x70); gotoxy(x,y); cprintf("%s",form);
return(ch2 == DOWN ? DOWN : UP);
}

```

```

int tgl_pupuk(void)
{
    int j=1;
    char ch2,ptm[4];

    gettext(15,5,65,22,buf2);
    sbbox(15,5,64,21,double_line,2);
    Status_OK(37,18,k_out[0],k_pupuk,0x70,2);
    textattr(0x2f);
    gotoxy(18,5); cprintf(" WAKTU PEMUPUKAN / BULAN ");
    textattr(0x7e); gotoxy(16,11);
    cprintf("      No.                Tanggal                ");
    textattr(0x20); gotoxy(18,8);
    cprintf("Lama Pemupukan   : [ %-4d ] detik",pupuk.intv);
    gotoxy(18,9); cprintf("Waktu pemupukan   : [ %s ]",pupuk.tm);
    gotoxy(20,13); cprintf("1.                [ %s ]",pupuk.tgl1);
    gotoxy(20,14); cprintf("2.                [ %s ]",pupuk.tgl2);
    gotoxy(20,15); cprintf("3.                [ %s ]",pupuk.tgl3);
    gotoxy(20,16); cprintf("4.                [ %s ]",pupuk.tgl4);
    cursor();
    for(;;)
    {
        switch(j)
        {
            case 1 : ch2 = inputnum(39,8,0x1e,0x20,&setp.intv); break;
            case 2 : ch2 = inputwkt(40,9,setp.tm,0); break;
            case 3 : ch2 = inputwkt(38,13,setp.tgl1,1); break;
            case 4 : ch2 = inputwkt(38,14,setp.tgl2,1); break;
            case 5 : ch2 = inputwkt(38,15,setp.tgl3,1); break;
            case 6 : ch2 = inputwkt(38,16,setp.tgl4,1); break;

```

```

    }
    if(ch2 == ALTO)
    { pupuk = setp; nocursor();puttext(15,5,65,22,buf2); return(1); }
    if(ch2 == ESC)
    { setp = pupuk; nocursor();puttext(15,5,65,22,buf2); return(0); }
    if(ch2 == DOWN || ch2 == ENTER) { j = ++j > 6 ? 1 : j ; }
    if(ch2 == UP) { j = --j < 1 ? 6 : j ; }
    }
}

```

```

int waktu_siram(void)
{
    int j=1;
    char ch2,ptm[4];

    gettext(15,5,60,21,buf2);
    sbbox(15,5,59,20,double_line,2);
    Status_OK(34,17,k_out[0],k_siram,0x70,2);
    textattr(0x2f);
    gotoxy(18,5); cprintf(" WAKTU PENYIRAMAN / HARI ");
    textattr(0x7e); gotoxy(16,10);
    cprintf("      No.              Waktu              ");
    textattr(0x20); gotoxy(18,8);
    cprintf("Lama Penyiraman : [ %-4d ] detik",siram.lama);
    gotoxy(20,12); cprintf("1.                [ %s ]",siram.tm1);
    gotoxy(20,13); cprintf("2.                [ %s ]",siram.tm2);
    gotoxy(20,14); cprintf("3.                [ %s ]",siram.tm3);
    cursor();
    for(;;)
    {
        switch(j)
        {
            case 1 : ch2 = inputnum(38,8,0x1e,0x20,&in.lama); break;
            case 2 : ch2 = inputwkt(36,12,in.tm1,0); break;
            case 3 : ch2 = inputwkt(36,13,in.tm2,0); break;
            case 4 : ch2 = inputwkt(36,14,in.tm3,0); break;
        }
        if(ch2 == ALTO)
        { siram = in; nocursor();puttext(15,5,60,21,buf2);return(1); }
        if(ch2 == ESC)
        { in = siram; nocursor();puttext(15,5,60,21,buf2); return(0); }
        if(ch2 == DOWN || ch2 == ENTER) { j = ++j > 4 ? 1 : j ; }
        if(ch2 == UP) { j = --j < 1 ? 4 : j ; }
    }
}

```

```

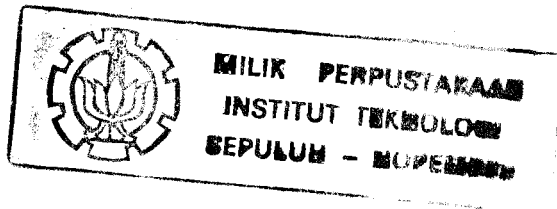
int device(void)
{
    int tc = 1, pilih1, pilih2, pilih3;
    char ch2;

    pilih1 = f.setfan; pilih2 = f.setsiram; pilih3 = f.setpupuk;
    sbbox(10,6,69,19,double_line,1);
    textattr(0x1e); gotoxy(15,6); cprintf(" SETTING PERALATAN ");
    textattr(0x70); gotoxy(19, 9);
    cprintf("   Fan                :   [%s]           ",cond[f.setfan]);
    gotoxy(19,11);
    cprintf("   Sistem Penyiraman   :   [%s]           ",cond1[f.setsiram]);
    gotoxy(19,13);
    cprintf("   Sistem Pemupukan    :   [%s]           ",cond1[f.setpupuk]);
    for(;;)
    {
        Status_OK(37,16,k_out[0],bottom1,0x30,1);
        switch(tc)
        {
            case 1 : ch2 = select(45, 9,cond,2,&pilih1,nothing); break;
            case 2 : ch2 = select(45,11,cond1,2,&pilih2,waktu_siram);
                     break;
            case 3 : ch2 = select(45,13,cond1,2,&pilih3,tgl_pupuk); break;
        }
        if(ch2 == ESC) { nocursor(); return; }
        if(ch2 == ALTO)
        {
            f.setfan = pilih1; f.setpupuk = pilih3; f.setsiram = pilih2;
            nocursor(); return;
        }
        if(ch2 == DOWN) { tc = ++tc > 3 ? 1 : tc ; }
        if(ch2 == UP) { tc = --tc < 1 ? 3 : tc ; }
    }
}

void All(void)
{
    struct tm *t;
    time_t now;
    char str[40];
    register int i;
    int ch,j;

    now = time(NULL);
    t = localtime(&now);
    sprintf(str,"%s, %d %s 19%d",hari[t->tm_wday],t->tm_mday,
            bulan[t->tm_mon],t->tm_year);
    gettext(1,2,80,23,buf2);
}

```



LAMPIRAN B-19

```
PrintMenu(1,25,bottom3,0x5f,14);
sbox(5,3,75,22,double_line,1);
for(i=0;i<70;i++) { gotoxy(6+i,7); cprintf("%c",'= '); }
tputc(40,7,'= '); tputc(5,7,'|| ');
tputc(75,7,'|| '); tputc(40,22,'_ ');
for(i=1;i<15;i++) { gotoxy(40,7+i); cprintf("%c",'|| '); }
textcolor(14);
gotoxy(35,3); cprintf(" MONITORING ");
gotoxy(17,7); cprintf(" PARAMETER ");
gotoxy(17,7); cprintf(" PERALATAN ");
gotoxy(20,5); cprintf(" Hari/Tgl : ");
textcolor(15);
gotoxy(34,5); cprintf("%s",str);
gotoxy(19,9); cprintf("[      ]");
gotoxy(8,12); cprintf("Temperatur");
gotoxy(8,15); cprintf("Kelembaban");
gotoxy(8,18); cprintf("Intensitas Cahaya");
gotoxy(43,9); cprintf("Sistem Pengontrol Temperatur");
gotoxy(43,12); cprintf("Sistem Pengontrol Kelembaban");
gotoxy(43,15); cprintf("Sistem Pengontrol Cahaya");
gotoxy(43,18); cprintf("Sistem Penyiraman ! Pemupukan");
textbackground(7);
for(i=0,j=0; i<3; i++,j+=3) space(8,13+j,30);
for(i=0,j=0; i<4; i++,j+=3) space(43,10+j,30);
while(ch != 27)
{
    textattr(0x74);
    gotoxy(10,9); cprintf("%d",tpwm);
    gotoxy(21,9); cprintf("%3d",z);
    if(awal)
    {
        gotoxy(10,13); cprintf("%-2d    ° Celcius",akhir->dat.temp);
        gotoxy(10,16); cprintf("%-2d    %",akhir->dat.humid);
        gotoxy(10,19); cprintf("%-4d    Lux",akhir->dat.light);
    }
    else
    {
        gotoxy(10,13); cprintf("%-2d    ° Celcius",pset.temp);
        gotoxy(10,16); cprintf("%-2d    %",pset.humid);
        gotoxy(10,19); cprintf("%-4d    Lux",pset.light);
    }
    gotoxy(45,10); cprintf("%s",cond[f.temp]);
    gotoxy(45,13); cprintf("%s",cond[f.humid]);
    gotoxy(45,16); cprintf("%s",cond[f.light]);
    gotoxy(45,19); cprintf("%s",cond[f.siram]);
    gotoxy(65,19); cprintf("%s",cond[f.pupuk]);
    if(bioskey(1) != 0) ch = bioskey(0) & 0x00ff;
}
```

```

    puttext(1,2,80,23,buf2);
}

addr *search(addr *node, int num)
{
    addr *info;
    int i = 0;

    info = node;
    while(info)
    {
        if(num == ++i || info == awal) return info;
        info = info->sbl;
    }
    return NULL;
}

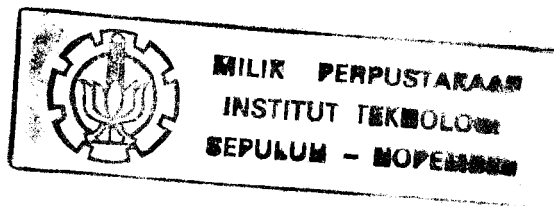
int print_data(int x, int y, addr *node, int mode,int attr)
{
    char *buf;

    if((buf = malloc(78)) == NULL) return;
    if(mode == 0)
        sprintf(buf," %13s      %2d      %2d      %2d      %2d
%2d      ", node->dat.time,node->dat.temp1,
node->dat.temp2,node->dat.temp3,node->dat.temp4,node->dat.temp);
    else if(mode == 1)
        sprintf(buf," %13s      %3d      %3d      %3d      %3d      %3d
%3d      %2d      ",
node->dat.time,node->dat.dry1,node->dat.wet1,node->dat.humid1,
node->dat.dry2,node->dat.wet2,node->dat.humid2,node->dat.humid);
    else
        sprintf(buf," %13s      %4d
", node->dat.time,node->dat.light);
    printstr(x,y,buf,attr);
    free(buf);
    return;
}

void win_select(int num,int mode)
{
    addr *info;
    int j = 0;

    textattr(0x1e); window(2,3,79,23);
    textattr(0x70); gotoxy(1,4); cprintf("%s",header[3*num]);
    gotoxy(1,5); cprintf("%s",header[3*num+1]);
    textattr(0x1e); gotoxy(25,2); cprintf("%s",header[3*num+2]);

```

LAMPIRAN B-21

```
window(1,1,80,25);
if(z < 17) info = awal;
else if(mode == RUN_MODE) info = search(akhir,16);
else info = search(br_atas,16);
while(info)
{
    j++;
    if(info == akhir) print_data(1,6+j,info,num,0x3e);
    else if(mode == STOP_MODE && j == pos-6)
        print_data(1,6+j,info,num,0x7e);
    else print_data(1,6+j,info,num,0x1e);
    if(mode == STOP_MODE && j == maks) break;
    info = info->ssd;
}
}

int shift_up(int num)
{
    if(dtop == awal) return;
    if(dtop != akhir) print_data(1,pos,dtop,num,0x1e);
    dtop = dtop->sbl;
    if(--pos < 7)
        { pos = 7; br_atas = br_atas->sbl; scroll(1,7,78,22,1,BAWAH); }
    print_data(1,pos,dtop,num,0x7e);
    return;
}

int shift_down(int num)
{
    if(dtop == akhir) return;
    print_data(1,pos,dtop,num,0x1e);
    dtop = dtop->ssd;
    if(++pos > 22)
        { pos = 22; br_atas = br_atas->ssd; scroll(1,7,78,22,1,ATAS); }
    else if(pos > maks+6) { maks++; br_atas = br_atas->ssd; }
    if(dtop == akhir) print_data(1,pos,dtop,num,0x3e);
    else print_data(1,pos,dtop,num,0x7e);
    return;
}

int Detail(void)
{
    char ch;
    int num,mode;

    gettext(1,1,80,25,buf2);
    PrintMenu(1,25,bottom2[0],0x5f,14);
    textattr(0x1f); dbox(1,2,80,24,double_line);
```

```

num = -1; mode = RUN_MODE;
for(;;)
{
    ch = readkey(1);
    switch(ch)
    {
        case F1 : num = 0; win_select(num,mode); break;
        case F2 : num = 1; win_select(num,mode); break;
        case F3 : num = 2; win_select(num,mode); break;
        case F4 : if(awal == NULL) break;
                    mode = !mode;
                    if(mode == RUN_MODE)
                    {
                        PrintMenu(1,25,bottom2[0],0x5f,14);
                        win_select(num,mode);
                    }
                    else
                    {
                        pos = z < 17 ? z+6 : 22; maks = pos-6;
                        dtop = br_atas = akhir;
                        PrintMenu(1,25,bottom2[1],0x5f,14);
                    }
                    break;
        case UP : if(mode == STOP_MODE) shift_up(num); break;
        case DOWN : if(mode == STOP_MODE) shift_down(num); break;
        case ESC : window(1,1,80,25); puttext(1,1,80,25,buf2);
                    return;
        default : if(num == -1 || mode == STOP_MODE) break;
                    if(f.param == TRUE)
                    {
                        if(z < 17)
                        {
                            if(akhir->sbl)
                                print_data(1,5+z,akhir->sbl,num,0x1e);
                            print_data(1,6+z,akhir,num,0x3e);
                        }
                        else
                        {
                            print_data(1,22,akhir->sbl,num,0x1e);
                            scroll(1,7,78,22,1,ATAS);
                            print_data(1,22,akhir,num,0x3e);
                        }
                        f.param = FALSE;
                    }
                    break;
    }
}
}
}

```

```

void about(void)
{
    char ch;

    gettext(15,6,65,19,buf1);
    sbbox(15,6,64,18,double_line,1);
    textattr(0x1e);
    gotoxy(20,8); cprintf("  GREEN HOUSE PROGRAMMABLE CONTROLLER ");
    gotoxy(20,10); cprintf("                FOR ORCHID                ");
    textattr(0x1f);
    gotoxy(20,12); cprintf("                Copyright (c) 1992 by                ");
    gotoxy(20,14); cprintf("  I Made Wira Satria ( 2882201077 ) ");
    Status_OK(37,16,k_out[0],k_top[0],0x70,1);
    do { ch = readkey(1); } while( ch != ALTO && tolower(ch) != 'o');
    puttext(15,6,65,19,buf1);
}

```

```

int menu(void)
{
    int pilih,p,p1,tt,no,cc;

    FillScr(178,0x3f,5);
    nocursor();
    PopMenu(2,1,top,5,1,0x5f,14);
    gotoxy(1,25); textattr(0x5f); cputs(k_top[1]);
    tt = 0; no = 0;
    while(f.end == FALSE)
    {
        pilih = TopMenu(top,5,k_top,0x5f,14,0x2f);
        switch(pilih)
        {
            case 0 : gettext(1,2,18,8,buf1);
                    p = VertMenu(2,2,nol,2,tt,k_nol,0x1f,14,0x2f);
                    puttext(1,2,18,8,buf1);
                    if(!p) about();
                    break;

            case 1 : gettext(6,2,28,9,buf1);
                    p = VertMenu(7,2,satu,4,tt,k_satu,0x1f,14,0x2f);
                    puttext(6,2,28,9,buf1); gettext(10,6,70,21,buf1);
                    switch(p)
                    {
                        case 0 : tmsample(); break;
                        case 1 : parameter(); break;
                        case 2 : device(); break;
                        case 3 : ConMode(); break;
                    }
                    puttext(10,6,70,20,buf1);

```

```

tt = p;
break;

case 2 : gettext(16,2,38,6,buf1);
xxx : p = VertMenu(17,2,dua,2,tt,k_dua,0x1f,14,0x2f);
switch(p)
{
    case 0 : gettext(18,4,30,8,buf2);
             p1 = VertMenu(19,4,dual,2,no,k_dual,
                          0x1f,14,0x2f);
             puttext(18,4,30,8,buf2);
             puttext(16,2,38,6,buf1);
             switch(p1)
             {
                 case 0 : All(); break;
                 case 1 : Detail(); break;
                 case 27 : tt = p; gettext(16,2,38,6,buf1);
                          goto xxx;
             }
             no = p1;
             break;

    case 1 : puttext(16,2,38,6,buf1);
             gettext(1,1,80,25,buf1);
             GraphView(); restorecrtmode();
             puttext(1,1,80,25,buf1);
             nocursor();
             break;

    default : puttext(16,2,38,6,buf1); break;
}
tt = p;
break;

case 3 : gettext(31,2,49,6,buf1);
yyy : p = VertMenu(32,2,tiga,3,tt,k_tiga,0x1f,14,0x2f);
switch(p)
{
    case 0 : gettext(33,4,58,8,buf2);
             p1 = VertMenu(34,4,tigal,3,no,k_tigal,
                          0x1f,14,0x2f);
             puttext(33,4,58,8,buf2);
             switch(p1)
             {
                 case 0 : ResetSystem(); break;
                 case 1 : f.setfan = FALSE; break;
                 case 2 : outportb(PB2,dev & 0x0b);
                          f.siram = 0; f.pupuk = 0;
             }
}

```

```

                                break;
                                case 27 : tt = p; goto yyy;
                                }
                                no = p1;
                                break;

                                case 1 : if(f.start == FALSE) f.start = TRUE;
                                else
                                    { t_error(err[1]); nocursor(); }
                                break;

                                case 2 : ResetSystem(); f.end = TRUE;
                                break;
                                }
                                puttext(31,2,49,6,buf1);
                                tt = p;
                                break;

                                case 4 : gettext(42,2,67,6,buf1);
                                p = VertMenu(43,2,empat,1,tt,k_empat,0x1f,14,0x2f);
                                puttext(42,2,67,6,buf1);
                                if(!p) FileInfo();
                                break;
                                }
                                }
                                textbackground(0); clrscr(); cursor(); return;
                                }

main()
{
    InitDevice();
    Init_var();
    oldhandler = getvect(8);
    setvect(8,ISR);
    menu();
    setvect(8,oldhandler);
    return 0;
}

```

RIWAYAT HIDUP



I Made Wira Satria Negara dilahirkan di Surabaya pada tanggal 9 Desember 1969, putra kedua dari Bapak I Made Widiana dan Ibu Tio Djie Him.

Terdaftar sebagai mahasiswa Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember Surabaya sejak tahun 1988 dengan nomor pokok 2882201077.

Pendidikan yang pernah ditempuh sebelumnya adalah :

- SD Mutiara, Singaraja, tahun 1976 - 1982.
- SMP Mutiara, Singaraja, tahun 1982 - 1985.
- SMA Negeri I, Singaraja, tahun 1985 - 1988.

Selama perkuliahan pernah menjadi asisten laboratoium Rangkaian Listrik, Elektronika dan Mikro.

